

UTILIZACIÓN DE HERRAMIENTAS DE SIMULACIÓN EN LA ROBÓTICA INDUSTRIAL



Diego Anerillas Aljama

Trabajo Final de Carrera
Ingeniería Técnica Informática de Sistemas

Escuela Politécnica Superior
Universidad de Lleida

Índice

1. Prólogo

- 1.1 Objetivos
- 1.2 Objeto de desarrollo.

2. La simulación en la robótica industrial.

- 2.1 Justificación del uso de software de simulación
- 2.2 Aplicación de la ingeniería simultánea
 - 2.2.1 Ingeniería concurrente vs. Ingeniería convencional
 - 2.2.2 Impacto cultural en la organización.
 - 2.2.3 Herramientas y técnicas asociadas a la ingeniería simultánea.

3. Análisis previo de los sistemas robóticos.

- 3.1 Universo del manipulador.
- 3.2 Características físicas del proceso.
- 3.3 Fundamentos matemáticos del movimiento del robot.
 - 3.3.1 Traslaciones y rotaciones.

4. Software de simulación

- 4.1 Justificación de elección Roboguide® de FANUC®
- 4.2 Programación de robot FANUC®
 - 4.2.1 Programación mediante TPE
 - 4.2.2 Programación en editor del software simulador

5. Simulación del sistema robótico

- 5.1 Diseño de la célula de trabajo.
- 5.2 Diagrama del proceso automático. Red de Petri.
- 5.3 Presentación del manipulador.
- 5.4 Implementación del sistema automático.
 - 5.4.1 Definición y configuración del robot y la controladora.
 - 5.4.2 Definición de objetos manipulados (PARTS)
 - 5.4.3 Definición y configuración de herramienta (UTOOL)
 - 5.4.4 Creación y configuración de partes fijas (FIXTURES)
 - 5.4.5 Definición de sistema de referencia usuario (UFRAME)
 - 5.4.6 Creación del programa.
 - 5.4.7 Simulación del proceso.

6. Ampliación de la célula de trabajo.

- 6.1 Rediseño de la célula robótica.
- 6.2 Configuración de la célula con robot adicional.
- 6.3 Creación del programa ampliado.
- 6.4 Simulación del sistema.

7. Evaluación de resultados y conclusiones.

8. Bibliografía

9. Anexos

10. Agradecimientos

Índice de figuras

La simulación en la robótica industrial.

Fig. 1. Equipo de trabajo multidisciplinario de ingeniería concurrente.

Fig. 2. Ciclo de vida del producto bajo el enfoque concurrente y tradicional.

Análisis previo de los sistemas robóticos.

Fig. 3. Tipos de sistemas de referencia.

Fig. 4. Rotación de sistema de referencia respecto al eje Z.

Fig. 5. Rotación respecto a un eje arbitrario a partir de ejes básicos.

Software de simulación

Fig. 6. Distintos modelos actuales de "Teach Pendant".

Fig. 7. Aspecto de consola "Teach Pendant".

Fig. 8. Estados de los indicadores "led" de la consola "Teach Pendant".

Fig. 9. Resumen de los botones de la consola "Teach Pendant".

Fig. 10. Ejemplo de programa en "display" TPE.

Fig. 11. Menús principales del dispositivo TPE

Fig. 12. Pantalla 1 de la creación de programa.

Fig. 13. Pantalla 2 de la creación de programa.

Fig. 14. Pantalla 1 de la definición de un punto.

Fig. 15. Estructura para la definición de un punto en la controladora robótica.

Fig. 16. Herramienta simple en TCP.

Fig. 17. Herramienta compleja en TCP.

Fig. 18. Pantalla 1 de la definición de herramienta.

Fig. 19. Pantalla 2 de la definición de herramienta.

Fig. 20. Definición de punto de origen de la herramienta.

Fig. 21. Pantalla de definición de origen de herramienta por el método de 3 puntos.

Fig. 22. Definición de Orient Origine Point de la herramienta.

Fig. 23. Definición de X Direction Point de la herramienta.

Fig. 24. Definición de Z Direction Point de la herramienta.

Fig. 25. Pantalla de definición de origen de herramienta por el método de 6 puntos.

Fig. 26. Pantalla de definición de origen de herramienta por el método entrada directa.

Fig. 27. Sistema de referencia definido por usuario (UFRAME).

Fig. 28. Pantalla 1 de los métodos de configuración de UFRAME.

Fig. 29. Pantalla 2 de los métodos de configuración de UFRAME.

Fig. 30. Pantalla de definición de UFRAME con el método de los 3 puntos.

Fig. 31. Resumen de registros periféricos de entrada.

Fig. 32. Resumen de registros periféricos de salida.

Fig. 33. Pantalla de configuración de registros periféricos I/O.

Fig. 34. Estado de configuración de registros periféricos de I/O.

Fig. 35. Resumen de la configuración del robot.

Fig. 36. Botones de ejecución directa de macros.

Fig. 37. Definición de límites de software fijos.

Fig. 38. Barras de acceso rápido del software HandlingPRO Roboguide®

Fig. 39. Imagen del menú "Cell Browser".

Fig. 40. Imágenes del menú "Navigator".

Fig. 41. Imagen del menú "Jog Coordinates Quick Bar".

Fig. 42. Imagen del menú "Teach Quick Bar".

Fig. 43. Imagen del menú "MoveTo Quick Bar".

Fig. 44. Imagen del menú "Target Tools".

Fig. 45. Imágenes de los menús "Draw features on parts".

Fig. 46. Imágenes de los menús "Teach Pendant".

Fig. 47. Imagen del menú "Robot Alarms".

Fig. 48. Imágenes de los menús "Robot Browser".

- Fig. 49. Imagen del menú "Run Panel".
- Fig. 50. Imagen del menú "Position Edit".
- Fig. 51. Imagen de la vista "View WireFrame".
- Fig. 52. Imagen del menú "Measure Tool".
- Fig. 53. Imagen del menú "Mouse Commands".
- Fig. 54. Imagen del menú principal "File".
- Fig. 55. Imagen del menú principal "Edit".
- Fig. 56. Imagen del menú principal "View".
- Fig. 57. Imagen del menú principal "Cell".
- Fig. 58. Imagen del menú principal "Robot".
- Fig. 59. Imagen del menú principal "View".
- Fig. 60. Imagen del menú "Teach Program".
- Fig. 61. Imagen del menú "Program Properties".
- Fig. 62. Imagen del menú principal "Test-run".
- Fig. 63. Imagen del menú principal "Test-run" > "Run Options".
- Fig. 64. Imagen del menú principal "Project".
- Fig. 65. Imagen del menú principal "Tools".
- Fig. 66. Imagen del menú principal "Window".
- Fig. 67. Imagen del menú principal "Help".

Simulación del sistema robótico

- Fig. 68. Diseño de la célula de trabajo del sistema robótico.
- Fig. 69. Diagrama de red de Petri. del sistema robótico.
- Fig. 70. Robot FANUC R2000iA Series.
- Fig. 71. Planos del robot FANUC R-2000iA/200F.
- Fig. 72. Asistente de creación de célula de trabajo "Step 1".
- Fig. 73. Asistente de creación de célula de trabajo "Step 2".
- Fig. 74. Asistente de creación de célula de trabajo "Step 3".
- Fig. 75. Asistente de creación de célula de trabajo "Step 4".
- Fig. 76. Asistente de creación de célula de trabajo "Step 5".
- Fig. 77. Asistente de creación de célula de trabajo "Step 6".
- Fig. 78. Asistente de creación de célula de trabajo "Step 7".
- Fig. 79. Asistente de creación de célula de trabajo "Step 8".
- Fig. 80. Inicialización de la controladora del robot.
- Fig. 81. Menú de las propiedades de configuración del robot R-2000iA/200F.
- Fig. 82. Menú de propiedades del objeto "CAJA" (parts).
- Fig. 83. Menú de propiedades del objeto "CAJA" (parts).
- Fig. 84. Definición de los objetos manipulados en la simulación (parts).
- Fig. 85. Menú "tooling" del software de simulación HandlingPRO®.
- Fig. 86. Definición de características en la pestaña "General".
- Fig. 87. Definición de características en la pestaña "UTOOL".
- Fig. 88. Definición de características en la pestaña "parts".
- Fig. 89. Definición de características en la pestaña "Simulation".
- Fig. 90. Creación de "fixtures" desde el menú "Cell Browser".
- Fig. 91. Configuración de la pestaña "General" de la parte fija "Soporte1".
- Fig. 92. Configuración de la pestaña "Calibration" de la parte fija "Soporte1".
- Fig. 93. Configuración de la pestaña "Parts" de la parte fija "Soporte1".
- Fig. 94. Configuración de la pestaña "Simulation" de la parte fija "Soporte1".
- Fig. 95. Configuración de los distintas pestañas de la parte fija "Base".
- Fig. 96. Configuración de los distintas pestañas de la parte fija "Fixture1".
- Fig. 97. Configuración de las partes fijas en el menú "Cell Browser".
- Fig. 98. Definición de UFRAME como nueva referencia.
- Fig. 99. Apariencia del menú "Teach Program".
- Fig. 100. Apariencia del botón "Record".
- Fig. 101. Apariencia del botón "Touchup".
- Fig. 102. Apariencia del botón "MoveTo".
- Fig. 103. Apariencia del botón "Forward".

- Fig. 104. Apariencia del botón "Backward".
- Fig. 105. Apariencia del botón "Inst".
- Fig. 106. Formato de instrucciones para definir posiciones.
- Fig. 107. Tipos de finalización de movimiento.
- Fig. 108. Definición de acción "Pickup".
- Fig. 109. Definición de acción "Drop".
- Fig. 110. Apariencia del menú "Current Position" del TPE.
- Fig. 111. Definición de posición para la aproximación al "Soporte1".
- Fig. 112. Definición de posición para la aproximación al "Soporte3".
- Fig. 113. Definición de posición para depositar "CAJA" en "Soporte2".
- Fig. 114. Definición de posición para capturar "Pieza1" en "Fixture2".
- Fig. 115. Definición de posición para la separación de base al depositar "Pieza1".
- Fig. 116. Definición de posición para capturar "Pieza1" de "Fixture1".

Ampliación de la célula de trabajo.

- Fig. 117. Utilización del menú "Run Panel" para la simulación.
- Fig. 118. Ampliación y rediseño de la célula de trabajo.
- Fig. 119. Red de Petri para la secuencia del "Robot1".
- Fig. 120. Red de Petri para la secuencia del "Robot2".
- Fig. 121. Menú de propiedades del objeto "Pieza2".
- Fig. 122. Menús de propiedades de la parte fija "SoporteA".
- Fig. 123. Configuración de objetos vinculados a la parte fija "BaseA".
- Fig. 124. Configuración de las opciones de simulación de la parte fija "BaseA".
- Fig. 125. Zona de colisión entre los manipuladores del sistema.
- Fig. 126. Definición de posición para la aproximación al "SoporteA".
- Fig. 127. Definición de posición para la aproximación a la "BaseA".
- Fig. 128. Definición de posición para "Pieza1" en "Base".
- Fig. 129. Imagen de la ampliación del sistema, implementado en "Prog1" y "Prog2".
- Fig. 130. Imagen de la secuencia de simulación de la célula de trabajo (1).
- Fig. 131. Imagen de la secuencia de simulación de la célula de trabajo (2).

1. Prólogo

1.1 Objetivos

Este trabajo final de carrera tiene como finalidad la utilización de una herramienta software de simulación para diseñar un sistema automático robotizado aplicado a un proceso de producción en un entorno industrial.

El objetivo principal es el estudio y el análisis de las capacidades y ventajas que ofrecen las herramientas actuales para la simulación "*offline*" en los entornos productivos frente a la programación realizada en el entorno real.

Para poder acometer este objetivo es necesaria la elección de una herramienta utilizada en la actualidad para llevar a cabo el caso práctico.

Para implementar el proceso se deben de asimilar y comprender los métodos de programación que admite la herramienta de simulación, además de las distintas funcionalidades y opciones de configuración de este software para realizar el proceso completo de forma virtual.

De esta forma se llevará a cabo el proceso real de una forma simulada, habiendo realizado previamente la configuración total del sistema: definición de herramientas del robot, especificación de sistemas de referencia, definición de piezas y objetos, definición de partes fijas, la definición de puntos, el trazado de las distintas trayectorias...

Se analizará el sistema robótico implementado mediante la simulación del proceso.

Se pretenderá realizar también una ampliación en el sistema implementado para evaluar la capacidad de inserción de cambios en la célula de trabajo mediante la utilización de la herramienta de simulación.

Se pretenden evaluar los resultados obtenidos una vez realizadas las simulaciones y así sacar las conclusiones pertinentes sobre este sistema virtual aplicado en un entorno real, así como sus posibles aplicaciones según el entorno de trabajo. Se propondrán también ampliaciones del trabajo y posibles mejoras con respecto a la aplicación del mismo en entornos reales.

Se analizará el concepto de ingeniería simultánea aplicado a los entornos industriales capaces de implementar en sus líneas de producción la simulación robótica.

Resumiendo, después de la realización de este TFC debemos ser capaces de afrontar la programación "*offline*" de una célula de trabajo aplicada a la producción de una planta industrial.

1.2 Objeto de desarrollo.

En concreto se pretende obtener un sistema robotizado de captura y posicionamiento de piezas (*pick and place*), de forma completamente automática realizando la totalidad del proceso mediante una herramienta de simulación para la programación del robot y la posterior evaluación del proceso. Se realizará un estudio básico de las condiciones físicas necesarias para realizar el proceso de "*pick and place*", teniendo en cuenta un entorno de trabajo real, según los diversos condicionantes que puedan afectar al sistema.

Una vez determinadas las precondiciones, elegiremos el robot a utilizar y los grados de libertad del mismo, para obtener la mayor eficiencia en las trayectorias de traslación de los objetos.

Para poder desarrollar el sistema especificado se utilizarán herramientas de simulación, en concreto el software Roboguide® de FANUC® y su librería de funciones para la simulación en el módulo HandlingPRO®, también conocido como SimPRO®, las cuales permitirán trabajar de forma "*offline*", aprovechando las ventajas que nos ofrece este método de trabajo con un manipulador robótico. Se implementará con esta herramienta el algoritmo de trabajo del manipulador conforme al controlador de un robot FANUC®, obteniendo de esta forma la simulación animada así como la rutina del robot para realizar este proceso de recogida y posicionamiento de las distintas piezas.

Podremos entonces visualizar el proceso con la herramienta de simulación y supervisarlos para reprogramarlos en caso de fallos o colisiones, incluso calibrarlos si tenemos conexión con el robot del entorno real.

2. Simulación en la robótica industrial.

2.1 Justificación del uso de software de simulación.

Cuando se realiza un estudio completo para la automatización de un proceso industrial es necesario el uso de la simulación del proceso mediante computadores para poder evaluar el sistema una vez diseñado y así evitar los posibles fallos y desajustes con el sistema robótico real montado. El principal motivo para realizar la simulación es principalmente la prevención de fallos y posibles daños en los mecanismos del brazo robótico y en el entorno de producción y así evitar costes derivados de la programación *"online"*.

Hoy en día, las herramientas software, diseñadas en su mayoría por los fabricantes de los propios robots industriales, nos permiten analizar el proceso a través de la simulación completa de las células robotizadas mediante computadores.

El concepto de simulación es posible gracias a la programación *"offline"* del sistema mediante el software específico. Las características de estas aplicaciones son tan avanzadas que nos permiten ajustar al máximo todos los parámetros de los movimientos y acciones de nuestro robot, incluso de trasladar esta programación directamente al robot y evitar la reprogramación de todo el proceso una vez montado físicamente el sistema en el entorno de producción.

Cabe mencionar que la interfaz para la programación *"offline"* de los sistemas robóticos es mucho más eficiente que la programación que podamos realizar mediante los controladores *"teach pendant"* o mediante un *"lead-by-the-nose"* para registrar los movimientos que el robot ha de realizar en el proceso determinado de forma *"online"*.

Para la programación *"offline"* actualmente es necesario un entorno gráfico en el software de programación y simulación, para poder visualizar el proceso habiendo definido previamente el "mundo" del entorno de trabajo del brazo. Gracias a la potencia de los ordenadores y las altas prestaciones de los sistemas de software de diseño asistido por ordenador (CAD) los sistemas utilizados para el modelado 3D son idóneos para los entornos de diseño y fabricación actuales, muy especialmente para los procesos industriales robotizados, como por ejemplo para la manipulación de objetos o piezas, procesos de soldadura, pintado, desbardado, ensamblaje de piezas, paletizado. En general podemos afirmar que cualquier proceso industrial que tenga un nivel de repetición dentro de una cadena de producción puede ser robotizado, beneficiándose de las ventajas que proporcionan los sistemas actuales.

Estas técnicas de programación conocidas como programación fuera de línea (offline programming OLP) utilizadas de forma adecuada nos permitirían realizar las siguientes tareas antes de trabajar en un entorno productivo:

- Desarrollar modelos, probarlos y optimizarlos, antes de ser utilizados en la fabricación de piezas y herramientas. Con ello se consigue analizar el comportamiento de sistemas muy complejos y difíciles de evaluar por otros procedimientos.
- Conocer cuál será el comportamiento de los sistemas antes de construirlos, sin perder de vista que los valores finales de los simuladores serán aproximaciones de los valores reales. Ello implica que los técnicos que realizan el diseño empleando simuladores deberán saber interpretar correctamente los resultados de la simulación en todo momento.
- Reprogramar el proceso fuera de una línea de fabricación que se encuentre ya en producción, si por alguna razón, cambian las necesidades. Con este método se reduce el tiempo de parada de las máquinas y robots.
- Anticipar el funcionamiento y puesta en servicio de las líneas de producción, ya que es un sistema independiente y puede realizarse en paralelo con el montaje de células robóticas.
- Mejorar la calidad y exactitud de los puntos de soldadura que podrían realizar los operarios, incluso los más expertos.
- Diseñar correctamente las trayectorias del elemento terminal (pinza o garra) así como sus velocidades y aceleraciones. Durante el modelado los programas simuladores informan de cualquier colisión o pérdida de proximidad entre los elementos del modelo y del entorno.

Pinza: herramienta utilizada para capturar y manipular objetos con una acción prensil en el proceso automatizado con robots.

Garra: herramienta fija utilizada para capturar objetos que serán manipulados en un proceso de producción industrial.

2.2 Aplicación de la ingeniería simultánea.

La Ingeniería Concurrente, también llamada Ingeniería Simultánea, Ingeniería Paralela, Ingeniería Total o Diseño Integrado de Producto, entre otros nombres, es una filosofía que incide directamente sobre la cultura de las organizaciones y replantea la forma convencional de trabajar los proyectos. Consideraremos algunas definiciones que detallan y resumen el concepto como tal.

El término de Ingeniería Concurrente surgió inicialmente en el verano de 1986 cuando fue utilizado en el reporte R-338 del Institute for Defense Analysis (IDA) de donde surge una de las definiciones más aceptadas universalmente:

"Un esfuerzo sistemático para un diseño integrado, concurrente del producto y de su correspondiente proceso de fabricación y servicio".

Pretende que los encargados del desarrollo desde un principio, tengan en cuenta todos los elementos del Ciclo de Vida del Producto (CVP), desde el diseño conceptual hasta su disponibilidad, incluyendo calidad, costo y necesidad de los usuarios.

"La ingeniería simultánea es el proyecto simultáneo de un producto y su proceso de manufactura".

"Es un enfoque integrado del desarrollo del producto que pone énfasis en las expectativas del cliente por medio de la fabricación de productos de alta calidad, con mayor rapidez y menor costo. Apoya los valores del trabajo multidisciplinario en equipo, como son la cooperación, la confianza, y el compartir e intercambiar los conocimientos y la información, de tal manera que la toma de decisiones durante la etapa del diseño proceda con énfasis en la consideración simultánea de todos los aspectos del ciclo de vida del producto".

Cabe destacar que la característica común en la aplicación de la Ingeniería Simultánea es que cada nuevo proyecto se maneja como una fuerza o equipo multidisciplinario trabajando tiempo completo. Pero además del enfoque de equipo multidisciplinario son esenciales también el uso de técnicas disciplinadas. Ninguno ofrece ganancias potenciales sin el otro.

Un grupo trabajando en diseño de producto o un comité extraído de distintos departamentos que se reúne regularmente no son Ingeniería Concurrente. Para que realmente se aplique Ingeniería Simultánea, el equipo de trabajo debe estar compuesto por ingenieros de diseño de producto, ingenieros de fabricación, personal de marketing, de compras, de finanzas y los principales proveedores del equipo de fabricación y componentes.



Fig. 1. Equipo de trabajo multidisciplinario de ingeniería concurrente.

Además debe tener carácter permanente durante toda la duración del proyecto para que su trabajo reciba la prioridad que requiere. La clave en los equipos de trabajo multidisciplinario está en que desde el inicio, cuando aún el diseño no es más que un bosquejo, los ingenieros de fabricación que forman parte del equipo tienen tanta información sobre el producto como cualquier otro miembro del grupo. Así pueden empezar a planificar las instalaciones de fabricación con el mismo concepto con el que los ingenieros de diseño están planificando el objeto que van a producir, es decir, se desarrolla un trabajo simultáneo.

Existe una interrelación directa y permanente que permite recomendaciones para reducir costes y número de piezas, elevando considerablemente la calidad. El hecho de tener personal de marketing en el equipo asegura que las metas de ventas sean alcanzables, y lo más importante es que este personal contribuye directamente a enfatizar en las expectativas de los clientes, lo cual permite dar una mayor ponderación a este aspecto en la ingeniería simultánea que en la ingeniería tradicional. Este enfoque permite identificar, en una fase suficientemente temprana, puntos donde deben hacerse rectificaciones, con un coste mucho menor que si se hicieran posteriormente. Como podemos comprobar, el concepto de ingeniería simultánea se adapta perfectamente a los entornos de producción programados mediante software de simulación, permitiendo reducir con mayor facilidad las correcciones del proceso y en definitiva el coste del ciclo de vida del producto.

2.2.1 Ingeniería Concurrente vs. Ingeniería Convencional

Se ha demostrado en proyectos realizados que el 80% del coste de manufactura se determina en la etapa de diseño, la cual es una de las primeras dentro del ciclo de vida del producto.

Esto determina la importancia definitiva que tiene la etapa de diseño y cómo las actividades que aquí se desarrollen pueden impactar en toda la organización. Para ilustrar el concepto, en la figura que se muestra a continuación se presenta el ciclo de vida de un proyecto de desarrollo de un producto bajo los dos enfoques: el de la ingeniería tradicional y el de la ingeniería simultánea, que se destaca notoriamente por la reducción del tiempo del ciclo de vida del producto (CVP).

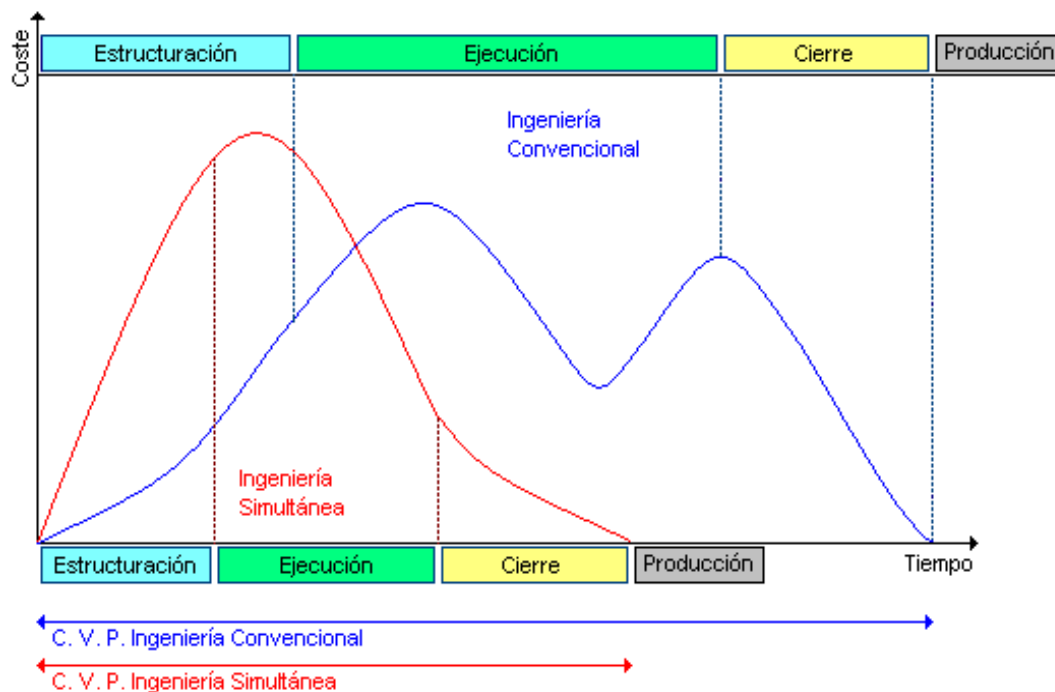


Fig. 2. Ciclo de vida del producto bajo el enfoque concurrente y tradicional.

Entre la bibliografía especializada señalamos el libro "Concurrent Engineering" de John R. Hartley, donde se muestra claramente lo que suele ocurrir tradicionalmente en las empresas dependiendo de su "cultura". La ingeniería convencional maneja un enfoque secuencial en el proceso de desarrollo de un producto. Según este enfoque cada área de la empresa, después de ejecutar la parte que le corresponde, transfiere su resultado al sector siguiente y queda a la espera. Cada unidad organizativa que recibe la información inevitablemente encontrará fallos según la perspectiva de su propia especialidad, y la devolverá al sector de origen para los ajustes correspondientes.

El sistema tradicional o compartimentado genera conflictos y trae como consecuencia muchos cambios y retroalimentaciones en las diferentes etapas, originados porque algunas características necesarias en las etapas posteriores no se consideraron desde el inicio del proceso, hecho que incide directamente en el incremento de los costos y el tiempo de desarrollo del producto. Utilizando la ingeniería tradicional se pone en riesgo la calidad del producto, ya sea porque no se tomaron las medidas correctivas o porque los cambios se aprecian como rectificaciones que no existirían si desde el inicio se hubiera trabajado en un diseño integrado de producto. En el gráfico podemos observar que en las primeras etapas el tiempo se incrementa, pero el tiempo total de ciclo se reduce sustancialmente. Un principio clave de la ingeniería concurrente es la introducción de la calidad desde el inicio del diseño, eliminando cualquier elemento que pueda verse afectado por variaciones en la producción. Se requiere de una cultura en la que cada una de las personas involucradas en el proceso sea responsable de la calidad. Es aquí donde encaja totalmente la filosofía de ingeniería concurrente y la de calidad total, cuyo fin común es la satisfacción del cliente.

2.2.2 Impacto cultural en la organización.

Asumir el reto de trabajar bajo el enfoque de ingeniería simultánea requiere tiempo y un esfuerzo especial de capacitación del talento humano, inicialmente para cambiar la forma de trabajo y posteriormente adiestrado en las técnicas y habilidades que se requieren para su implementación. Inicialmente puede haber resistencia y conflictos por parte de las personas, especialmente por los jefes de área que sienten la pérdida de poder a favor de los equipos de proyectos, o también por la presión de los plazos de entrega de tareas que anteriormente se realizaban después de meses, y con el enfoque simultáneo muchas veces se deben tener en pocos días.

Definitivamente no es fácil cuando se ha utilizado la ingeniería secuencial con etapas bien definidas, donde cada departamento tiene responsabilidad de una función en particular y ésta se desarrolla después de que la anterior ha sido completada.

Los sistemas jerárquicos que tienen la mayoría de las organizaciones dificultan el trabajo en equipo, por lo cual se debe ser cuidadoso y prepararse para un proceso de transición hacia estructuras organizativas más flexibles y dinámicas, que faciliten y apoyen el desarrollo de nuevas actitudes valorando y promoviendo el trabajo en equipo, así como también hacia sistemas de evaluación de desempeño donde se tenga en cuenta no sólo la competencia técnica, sino también la creatividad y el éxito como integrantes de un equipo.

Algunas veces la tendencia es una reorganización con énfasis en el mercado y la competitividad, como en el caso de las estructuras circulares, las matriciales y las unidades estratégicas de negocios.

Los paradigmas actuales que rigen tanto la formación de los profesionales como el desempeño del talento humano en las empresas tienden a valorar el trabajo individual. La ingeniería concurrente requiere de un cambio cultural en el que la creatividad, la polivalencia, el trabajo multidisciplinario, la confianza y el entusiasmo sean la razón de ser de la concepción del trabajo. En muchas empresas de clase mundial, especialmente las japonesas, el enfoque simultáneo está tan arraigado en su cultura que el trabajo en equipo es inherente a las decisiones políticas y a las operaciones del día a día y, por supuesto, no conciben una forma diferente de trabajar, es simplemente sentido común.

2.2.3 Herramientas y técnicas asociadas a la Ingeniería Simultánea.

Aunque para aplicar los fundamentos de ingeniería concurrente no es estrictamente necesario que la empresa haga fuertes inversiones en computadoras, software sofisticado y equipos de última generación, no cabe duda de que sería ideal y de gran importancia disponer de herramientas y técnicas que faciliten y agilicen el proceso de toma de decisiones, así como el de cambio cultural. El desafío para fabricar hoy es apoyarse en técnicas que queden una ventaja competitiva por el uso de tecnología de información y herramientas que permitan el enlace eficiente a través de toda la cadena de valor.

Sin embargo, mientras se sigan empleando las técnicas de gestión tradicionales no se puede decir que se está aplicando ingeniería concurrente. El liderazgo tiene que venir de arriba, con altos directivos apoyando decididamente y facilitando los medios para desarrollar las capacidades de los equipos de trabajo. Además deben practicar una política de delegación, permitiendo que el equipo saque adelante el proyecto con mucha autonomía, hasta el punto que si un alto directivo cree realmente que el proyecto marcha en dirección equivocada, debe reunir a todo el equipo de trabajo para llegar a una verdadera decisión de consenso.

Entre las herramientas esenciales para la Ingeniería Simultánea se incluyen los Sistemas de Diseño Asistido por Computador (CAD), que son muy importantes porque permiten hacer simulación en paralelo, con lo cual se disminuye el riesgo en el momento de decidir por la opción más práctica. Para maximizar los beneficios de la Ingeniería Concurrente se debe tender hacia la utilización no sólo del Diseño sino también de su integración con la Fabricación Asistida por Computador (CAD/CAM).

Con una correcta combinación de "*hardware*" y "*software*" que permita a ingenieros de diversas disciplinas trabajar en paralelo se logra disminuir la fabricación de prototipos y sus plazos de ejecución pueden reducirse considerablemente. Igualmente, la Ingeniería Asistida por Computador (CAE) es otra de las herramientas que permite a través de software elaborar proyectos más eficientemente. Es importante hacer referencia que el uso de CAO/CAM/CAE tiene como prioridad automatizar el trabajo de la elaboración de proyectos, mientras que la Ingeniería Simultánea se ocupa más de asegurar la interacción entre los miembros del equipo que trabajan en el proyecto.

El enfoque simultáneo se apoya en técnicas como:

- Despliegue de la función de calidad (QFD). Se especifica el producto en una matriz, relacionando los deseos del consumidor (atributos del cliente) con las características cuantificadas de ingeniería.
- Control estadístico de proceso (SPC). Conjunto de técnicas y procedimientos aplicados a las diversas fases del proceso de manufactura para reducir o eliminar las fallas en la calidad del producto final.
- Análisis de fallos (FMEA). Conjunto de actividades que identifican las posibles fallas de un producto o proceso y sus causas, las medidas que pueden impedir o reducir la posibilidad de ocurrencia y la documentación del proceso, cuyo resultado será la recomendación de mejoras.
- Diseño para manufactura y ensamble (DFMA). A través de software se alerta al diseñador de productos sobre las implicaciones de su trabajo en la fase de manufactura.
- Los métodos de Taguchi. Conducen a un diseño robusto no afectado por las variables del proceso de producción.
- Justo a tiempo (JIT). Método de producción que trata de tener disponibles los materiales únicamente cuando se requieren, reduciendo considerablemente los costos de inventario.
- Benchmarking. Conjunto de procedimientos a través de los cuales se comparan parámetros y especificaciones de un producto con los de la competencia, que tiene el máximo desempeño.
- Manufactura integrada por computador. Uso de software que permite aprovechar los recursos de la informática para conectar el equipo de manufactura con la base de datos del área de proyectos.

La tendencia actual es a buscar mayor integración funcional. Idealmente, la ingeniería simultánea busca aplicar las herramientas que permitan a todos los integrantes del equipo del proyecto tener acceso compartido a la información actualizada del mismo, de tal manera que puedan almacenarlos y procesarlos de forma transparente. La fábrica del futuro deberá estar, en su concepción ideal, libre de barreras organizativas y geográficas, con algunas limitantes impuestas por la dinámica de los negocios.

Afortunadamente, la Ingeniería Simultánea se presta a una introducción gradual y es tan útil en los proyectos pequeños como en los grandes. Por tanto, las pequeñas empresas como las grandes pueden seleccionar los elementos que necesitan.

Por otra parte, en las compañías de menor tamaño es muy probable que utilicen aisladamente algunos elementos típicos de ingeniería concurrente, simplemente porque a menudo los empleados deben asumir más de una responsabilidad. Algunos sectores se han interesado por esta filosofía, tal es el caso del sector metal-mecánico, que ha encaminado esfuerzos al conocimiento del tema. Cada día serán más las empresas que tomarán en consideración la forma de trabajo simultánea, sobretudo en el campo de la producción automatizada.

Resumiendo, la ingeniería simultánea permite a las empresas incrementar la productividad ante una reducción significativa de los tiempos de entrega y de los costos evitando rectificaciones del producto, así como una integración entre los diferentes departamentos o áreas y mejoras en la comunicación de los mismos. Además, permite mejorar la calidad del producto, debido a que desde su misma concepción y diseño se prevé las características y condiciones en las cuales se desarrollará, de tal manera que se condiciona el producto a las especificaciones del cliente desde el inicio del ciclo de vida del producto. Por esto, la Ingeniería Concurrente conlleva excelentes resultados que permiten, a las empresas que la apliquen, alcanzar verdaderas ventajas competitivas.

Hay que tener muy en cuenta que la ingeniería simultánea se debe adaptar a las características de cada organización, con el fin de crear un ambiente y una estructura para su aplicación de forma eficiente.

3. Análisis previo de los sistemas robóticos.

3.1 Universo del manipulador.

"Un robot industrial es un manipulador multifuncional reprogramable diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales, mediante movimientos variables programados para la ejecución de una diversidad de tareas".

Robotics Industries Association (RIA)

El proceso de planificación de un proceso industrial para la producción en cadena, o cualquier tipo de actividad que tenga una función determinada en un sistema automático, puede plantearse como una acción automática ejecutada por un manipulador robótico industrial.

Así mismo, debemos considerar que cada acción dentro de una cadena de producción automatizada tiene una serie de condiciones que el robot manipulador debe cumplir.

Por ello los robots que se utilizan en los procesos industriales son analizados para la tarea que han de desempeñar según el siguiente criterio de clasificación:

- *Grado de libertad:* Es el número de ejes con los que el robot cuenta para poder realizar sus movimientos. Dos ejes son requeridos para llegar a cualquier punto en un plano, tres ejes son necesarios para llegar a cualquier punto del espacio de un sistema acotado. Para controlar completamente la orientación del extremo del brazo (es decir, la muñeca), tres ejes más (guiñada, cabeceo y balanceo) son obligatorios. Algunos diseños (por ejemplo, el robot SCARA) intercambian limitaciones en las posibilidades de movimiento en el coste, velocidad y precisión.
- *Grados de libertad:* es la misma clasificación que el número de ejes.
- *Dotación de trabajo:* la región del espacio un robot puede alcanzar.
- *Cinemática:* la disposición real de los miembros rígidos y las articulaciones del robot, que determina los movimientos posibles del robot. Las clases de la cinemática del robot son articulados, cartesiano, paralelo y SCARA.
- *La capacidad de carga o carga útil:* es peso que puede levantar un robot.

- *Velocidad*: la rapidez con que el robot puede colocar el extremo de su brazo. Esta puede ser definida en términos de velocidad angular o lineal de cada eje, o una velocidad compuesta es decir, la velocidad del extremo del brazo cuando todos los ejes se mueven.
- *Aceleración*: es la capacidad con la que un eje puede ser acelerado. Este es un factor limitante, ya que un robot no puede alcanzar su velocidad máxima especificada para los movimientos de corta distancia o un camino complejo que requiere frecuentes cambios de dirección.
- *Precisión*: es la capacidad que tiene un robot para alcanzar una posición exacta determinada. Cuando la posición absoluta del robot se mide y se compara con la posición definida, el error es una medida de precisión. La precisión se puede mejorar con detección externa, por ejemplo, un sistema de visión o IR. La precisión puede variar con la velocidad y la posición dentro de la dotación de trabajo y con la carga útil.
- *Repetibilidad*: capacidad del robot para volver a una posición programada. No se considera lo mismo que la precisión. Puede ser que cuando el robot adopte una cierta posición x, y, z sólo obtenga una precisión de 1 mm de esa posición. Esta sería la precisión que se puede mejorar mediante la calibración. Si esa posición se programa en la memoria del controlador y cada vez que se envía allí vuelve a 0,1 mm de la posición programada entonces la repetibilidad estará dentro de 0,1 mm.
- *Control de movimiento*: para algunas aplicaciones, tales como el montaje "pick and place", el robot sólo necesita volver repetidamente un número limitado de veces a unas posiciones predefinidas. Para aplicaciones más sofisticadas, como la soldadura y el acabado (pintura por "spray"), el movimiento debe ser continuamente controlado para seguir un camino en el espacio, con la orientación y la velocidad controlada.
- *Fuente de alimentación*: algunos robots usan motores eléctricos, otros utilizan actuadores hidráulicos. Los primeros son más rápidos, y estos últimos son más fuertes y apropiados en aplicaciones tales como pintura a pistola, donde una chispa podría provocar una explosión, sin embargo, bajo el aire de presurización interna del brazo puede evitar la entrada de vapores inflamables, así como otros contaminantes.
- *Transmisión*: algunos robots utilizan los motores eléctricos en las articulaciones a través de engranajes y otros utilizan el motor a la articulación directa (transmisión directa). Los brazos de robot pequeños son capaces de trabajar a alta velocidad, bajo pares motor de corriente continua, que generalmente requieren un sistema de engranajes complejo. En estos casos, la transmisión armónica es de uso frecuente para evitar desajustes entre engranajes.

- *Conformidad*: se trata de la medida del ángulo o la distancia que un eje del robot se mueve cuando se aplica una fuerza. La conformidad cuando un robot va a una posición transportando su carga útil máxima será en una posición ligeramente inferior a la que está llevando a ninguna carga. El cumplimiento también pueden ser responsable de desajuste transportando cargas pesadas en cuyo caso la aceleración tendría que ser reducida.

3.2 Características físicas del proceso.

El proceso industrial que se va a realizar es un sistema manipulación de de "pick and place" situado en un entorno de una planta de producción industrial.

El manipulador tendrá como objetivo trasladar piezas y materiales de un lugar a otro dentro de su rango de alcance. Mediante sistemas accesorios automáticos (como cintas transportadoras y otros robots manipuladores) se irá dotando al robot de piezas para que éste pueda manipular, capturándolas desde su posición inicial y depositándolas en la posición final deseada. Las piezas que intervendrán en el proceso se definirán sobre soportes, manteniendo siempre la misma posición de origen, y gracias a la programación del robot manipulador, mantendrán siempre la mismas posiciones de destino.

En cuanto a los objetos manipulados, serán piezas y herramientas cuya masa no superará los 20Kg, y la distancia entre soportes no superará los 2,5 metros de longitud. Para poder manipular las piezas mencionadas será necesario dotar al brazo robótico de una herramienta EOAT (End of Arm Tool) capaz de coger dichas piezas para poder transportarlas de su posición de origen a su posición de destino. Para ello se utilizará una pinza con una capacidad prensil con la capacidad de sujetar una masa de 20 kg, que es la masa máxima que ha de manipular.

Teniendo en cuenta estas características mediante el cálculo del "payload" (carga útil incluyendo el peso de la muñeca robótica, la EOAT y la pieza) hemos de seleccionar un robot con un "payload" y rango de alcance suficientes para poder acometer las necesidades planteadas. El robot elegido debe de tener los grados de libertad suficientes como para permitir los movimientos de traslación en la célula de trabajo.

Hemos de tener en cuenta que las piezas deberán ser volteadas mientras se trasladan de posición, lo que implicará que la libertad de movimiento del robot ha de ser elevada.

3.3 Fundamentos matemáticos del movimiento del robot..

Sabemos que la posición de un punto en el espacio euclídeo tridimensional viene determinada por tres cantidades, que llamamos sus coordenadas, y decimos que están expresadas en algún sistema de referencia, formado por tres ejes, usualmente rectilíneos.

En lo sucesivo usaremos exclusivamente sistemas de referencia rectilíneos, ortogonales (es decir, con sus tres ejes perpendiculares dos a dos), normalizados (es decir, las longitudes de los vectores básicos de cada eje son iguales) y dextrógiros (el tercer eje es producto vectorial de los otros dos).

Usaremos, pues, simplemente el término "sistema" para referirnos a sistemas ortonormales.

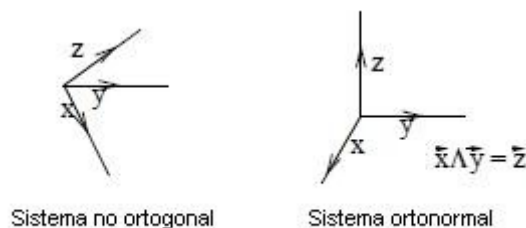


Fig. 3. Tipos de sistemas de referencia.

Las coordenadas de un punto, denotadas por $(x; y; z)$, son las proyecciones de dicho punto perpendicularmente a cada eje, o, equivalentemente, las componentes del vector que lo une al origen de coordenadas. En lugar de usar éstas, nos será más conveniente el uso de las llamadas coordenadas homogéneas, en la forma:

$$\begin{pmatrix} x' \\ y' \\ z' \\ w \end{pmatrix} \quad \text{donde} \quad \begin{aligned} x' &= xw \\ y' &= yw \\ z' &= zw \end{aligned}$$

siendo w una cantidad arbitraria, que se suele tomar como 1. Si, como resultado de algún cálculo, w fuese distinto de 1, las coordenadas usuales se reconstruyen simplemente dividiendo las tres primeras coordenadas homogéneas entre esta cuarta.

3.3.1 Traslaciones y rotaciones.

La traslación de un punto x por un vector v es obviamente el punto x' tal que:

$$\vec{x}' = \vec{x} + \vec{v}.$$

Esta expresión puede escribirse como:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

Pero también como el producto de una matriz por un vector homogéneo, en la forma:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Esto tiene la ventaja de que, si: $\vec{x}' = H \cdot \vec{x}$ entonces $\vec{x} = (H)^{-1} \cdot \vec{x}'$

donde se puede calcular la inversa, que resulta ser:

$$(H)^{-1} = \begin{pmatrix} 1 & 0 & 0 & -v_x \\ 0 & 1 & 0 & -v_y \\ 0 & 0 & 1 & -v_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

lo cual es consistente con el hecho de que x está trasladado por un vector $-v$ respecto a x' .

Respecto a la rotación alrededor de un eje, en el caso bidimensional, se está rotando con respecto a un eje z perpendicular al plano de la figura.

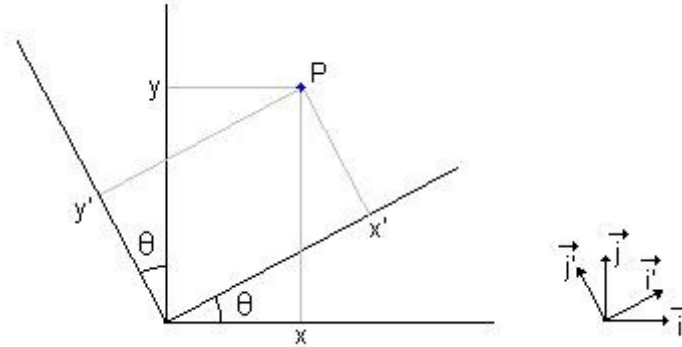


Fig. 4. Rotación de sistema de referencia respecto al eje Z.

Llamando i, j a los vectores básicos del sistema original, e i' y j' a los del sistema girado, se tiene que

$$\vec{x} = x\vec{i} + y\vec{j} = x'\vec{i}' + y'\vec{j}'$$

y también

$$\begin{aligned}\vec{i}' &= \cos \theta \vec{i} + \sin \theta \vec{j} \\ \vec{j}' &= -\sin \theta \vec{i} + \cos \theta \vec{j}\end{aligned}$$

Es decir que $x\vec{i} + y\vec{j} = x'(\cos \theta \vec{i} + \sin \theta \vec{j}) + y'(-\sin \theta \vec{i} + \cos \theta \vec{j})$

Igualando componente a componente, escribimos la matriz como

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Si generalizamos a tres dimensiones, como la coordenada z no varía y la cuarta coordenada homogénea sigue siendo 1, tenemos

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

Para hallar la transformación inversa basta ver que desde el punto de vista de R' , R está rotado un ángulo $-\theta$

luego podemos afirmar que

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) & 0 & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

Esta operación es más simple que invertir la matriz, aunque por supuesto, equivalente. En general, si hubiéramos rotado alrededor de otro de los ejes básicos, se puede ver que

$$Rot(x, \theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rot(y, \theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Rot(z, \theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Cabe destacar el cambio de signo en la rotación alrededor del eje y , debido a que, si el eje alrededor del cual rotamos nos apunta, los otros dos forman un ángulo de 90° en el caso de x y z , pero de -90° en el caso de y . Se pueden aplicar a un punto tantas transformaciones sucesivas (rotaciones y traslaciones) como se quiera. La operación resultante vendría dada por una matriz que sería producto de las matrices de cada operación, aplicadas en el orden correcto, dado que el producto de matrices no es conmutativo.

Se pone más a la derecha la primera transformación que se aplique, siendo expresado como

$$Y = T_2 R_3 T_1 R_2 R_1 X$$

Significa que se aplica al punto X la rotación 1, seguida de la rotación 2, seguida de la traslación 1, luego la rotación 3 y por último la traslación 2.

Veamos ahora cuál sería la matriz de rotación respecto a un eje cualquiera. Sea un eje que pasa por el origen definido por un vector unitario alrededor del cual giraremos un ángulo θ .

$$\vec{r} = (r_x, r_y, r_z)$$

Esta rotación se podrá descomponer en tres rotaciones sobre los ejes básicos, lo que equivaldrá a:

- Rotar un ángulo α alrededor de x , con lo que P pasará a la posición P' .
- Rotar un ángulo $-\beta$ alrededor de y , con lo que P' pasará a la posición P'' .
- Rotar un ángulo θ alrededor de z , que es la rotación que se pide.
- Rotar un ángulo β alrededor de y , deshaciendo la segunda rotación
- Rotar un ángulo $-\alpha$ alrededor de x , deshaciendo la primera rotación.

Entonces tenemos que $R_{\vec{r},\theta} = R_{x,-\alpha} R_{y,\beta} R_{z,\theta} R_{y,-\beta} R_{x,\alpha}$

o también

$$\begin{aligned} \text{El } R_{\vec{r},\theta} = & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & s\alpha & 0 \\ 0 & -s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\beta & 0 & s\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ & \cdot \begin{pmatrix} c\beta & 0 & -s\beta & 0 \\ 0 & 1 & 0 & 0 \\ s\beta & 0 & c\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

ejemplo tratado se resume en la siguiente figura:

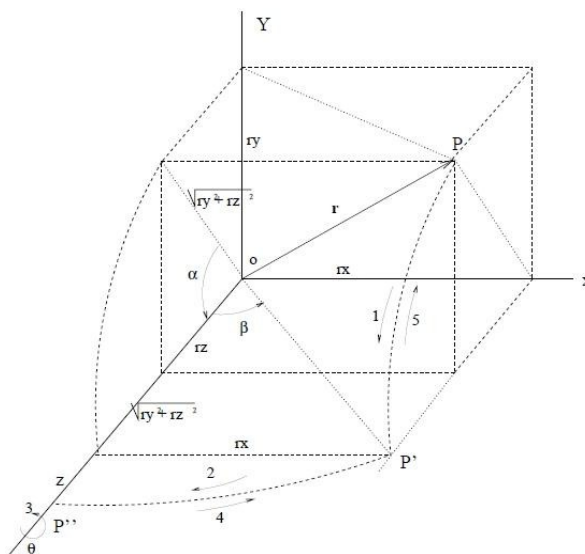


Fig. 5. Rotación respecto a un eje arbitrario a partir de ejes básicos.

Podemos comprobar que se cumple

$$\begin{aligned}
 s\alpha &= \frac{r_y}{\sqrt{r_y^2 + r_z^2}} \\
 c\alpha &= \frac{r_z}{\sqrt{r_y^2 + r_z^2}} \\
 s\beta &= \frac{r_x}{|r|} = r_x \\
 c\beta &= \frac{\sqrt{r_y^2 + r_z^2}}{|r|} = \sqrt{r_y^2 + r_z^2}
 \end{aligned}$$

Al multiplicarlo todo queda

$$R_{\vec{r},\theta} = \begin{pmatrix} r_x^2 v\theta + c\theta & r_x r_y v\theta - r_z s\theta & r_x r_z v\theta + r_y c\theta & 0 \\ r_x r_y v\theta + r_z s\theta & r_y^2 v\theta + c\theta & r_y r_z v\theta - r_x s\theta & 0 \\ r_x r_z v\theta - r_y s\theta & r_y r_z v\theta + r_x s\theta & r_x^2 v\theta + c\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

siendo $s\theta = \sin \theta$, $c\theta = \cos \theta$ y $v\theta = 1 - \cos \theta$

Evidentemente, si se hace $r_x = r_y = 0$ se obtiene la rotación alrededor del eje z.

De la misma forma podemos obtener la rotación de los otros dos ejes aplicando este mismo método.

Debemos destacar que cualquier secuencia consecutiva de transformaciones se puede especificar de dos formas:

- Realizando la rotación que lleva un sistema al otro, alrededor de los ejes iniciales.
- Realizando la rotación que lleva un sistema al otro alrededor de uno de los ejes girados, es decir, los que resultaron de la última transformación.

En el primer caso, la matriz que describe esta transformación deberá *pre-multiplicarse* por la que describía las transformaciones efectuadas hasta el momento, obteniendo la transformación total.

En el segundo caso la matriz que describe esta transformación deberá *post-multiplicarse* por la que describía las transformaciones efectuadas hasta el momento, obteniendo la transformación total.

4. Software de simulación.

4.1 Justificación de elección Roboguide de FANUC

El mercado actual en el ámbito de la robótica industrial ofrece diversas posibilidades en cuanto a la elección de una herramienta de simulación, generalmente asociada a una marca de robots concreta. La gran mayoría de las herramientas de simulación comerciales están desarrolladas por la propia marca del robot manipulador con el que se pretende trabajar, teniendo en cuenta que los fabricantes de robots proveen el software de simulación adaptado a sus modelos de controladoras. Este hecho tiene cierta lógica ya que el fabricante es el proveedor de la controladora y es el que mejor conoce todas las funcionalidades que ofrece su dispositivo a la hora de trasladarlas a un software de estas características.

Dadas las grandes ventajas que ofrece el sistema de simulación de un proceso de forma "offline" los fabricantes han ampliado su oferta comercial ofreciendo estas herramientas a sus clientes, para poder mejorar el sistema de programación de sus manipuladores y para ampliar también su oferta de productos con este software tan potente.

Después de revisar los distintos fabricantes de robots industriales actuales (ABB, FANUC, KIKI,...) hemos podido observar que la herramienta Roboguide® de FANUC® es de las más completas. Dada la relevancia de la marca FANUC en el entorno industrial nos hemos decantado por su herramienta, teniendo en cuenta la aplicación real del proceso industrial a simular en el entorno industrial actual. Roboguide es un potente software de programación "offline", que nos permite supervisar en todo momento el proceso programado evitando los costes de programación y pruebas en un entorno real.

El interfaz que la aplicación ofrece nos permite diseñar el entorno de la célula de una forma muy completa y con cierta facilidad, además de permitir configurar cada uno de los objetos que intervendrán en el proceso de producción. El objetivo final que nos permite alcanzar es la verificación de la operación completa del robot. Podemos detectar fácilmente las colisiones o interferencias producidas entre el robot y el resto de objetos de la célula. Nos permite importar archivos CAD para poder incluir piezas ya existentes en el entorno virtual. Además incluye un "Teach Pendant" virtual integrado que simula las consolas reales.

Una de las características destacadas de la aplicación es que es capaz de crear programas de referencia automáticamente con el robot real para poder calibrar el sistema robótico, siempre que esté disponible la conexión con el robot real.

El módulo HandlingPRO de Roboguide nos permite realizar un sistema de manipulación "Pick and place" con los elementos utilizados con mayor frecuencia para este tipo de procesos, lo que se adapta perfectamente con el proceso que se precisa programar y simular.

4.2 Programación del robot FANUC.

La programación de un robot podría definirse como la ejecución de un conjunto de comandos y órdenes de forma secuencial para que, paso a paso, irán ejecutando cada una de las órdenes que completarán los distintos procesos de nuestro sistema.

Disponemos de dos métodos posibles para poder diseñar el programa del manipulador mediante la herramienta de simulación:

- *"Roboguide Virtual Teach Pendant Editor"*. La aplicación nos ofrece la posibilidad de programar el robot con un *"teach pendant"* virtual, capaz de trabajar con todas las funcionalidades del manipulador administradas por la controladora. El *"teach pendant"* virtual tiene el mismo software que un *"teach pendant"* real conectado físicamente al manipulador. Se utiliza para crear instrucciones complejas con las funcionalidades de los programas TP.
- *"Roboguide Simulation Programs and editor"*. La aplicación también nos ofrece un editor intuitivo y muy completo, que permite la programación de programas mediante un entorno gráfico. Las grandes ventajas de este editor son que permite crear programas robóticos de una forma más cómoda que en un terminal TPE y ofrece la gran ventaja de evaluar de forma *"offline"* los ciclos de los procesos de las células de trabajo mediante las animaciones generadas por el programa de simulación, controlando los diversos factores que pueden interferir en nuestra simulación sin tener que sufrirlos en un entorno real (trayectorias incorrectas o imposibles, colisiones, puntos fuera de alcance...).

4.2.1 Programación TPE

Para la realización de la definición del entorno de la célula de trabajo, lo que permitirá el funcionamiento del manipulador robótico, utilizaremos el Teach Pendant virtual, tal y como se realizaría en las líneas de producción reales. También contrastaremos el proceso de implementación con el editor de simulación que nos ofrece el software.

Comenzaremos con una breve descripción de la utilización de este interfaz, así como de sus principales funcionalidades, menús y teclas. Existen distintos modelos en el mercado dependiendo del modelo de robot y de la controladora en cuestión.

El teach pendant de los robots FANUC tiene el siguiente aspecto:



Fig. 6. Distintos modelos actuales de "Teach Pendant".

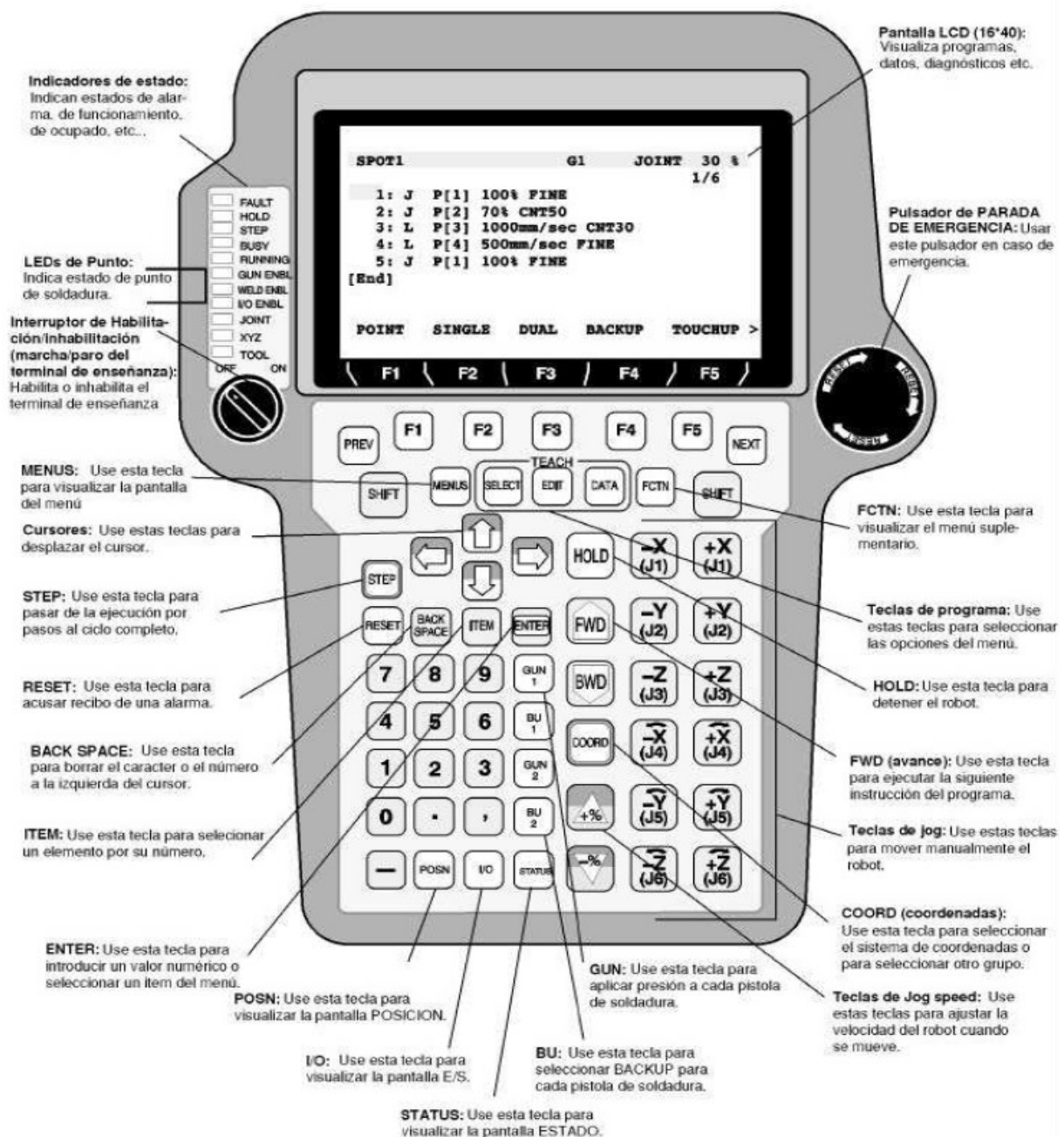


Fig. 7. Aspecto de consola "Teach Pendant".

El esquema superior indica las funcionalidades estándar asociadas a los botones del TP de FANUC. Existen algunas diferencias entre los distintos modelos de controladoras, aunque las diferencias son mínimas.



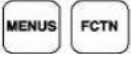




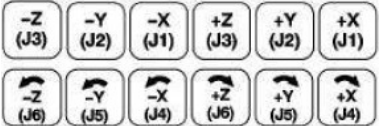







La tabla siguiente especifica cada uno de los estados según el cuadro de leds superior izquierdo:

LED	FUNCION
FAULT	FAULT LED indica que existe una alarma.
HOLD	HOLD LED indica que se pulsa el botón HOLD o que se recibe una señal HOLD.
STEP	STEP LED indica que se está bajo el modo de operación paso a paso.
BUSY	BUSY LED se enciende mientras el robot está trabajando. También se enciende cuando la CPU realiza trabajos relacionados con el movimiento del robot (copiar, pegar, imprimir...)
RUNNING	RUNNING LED indica que se está ejecutando el programa.
GUN ENBL	GUN ENBL indica que se puede abrir o cerrar la pinza.
WELD ENBL	WELD ENBL indica que puede realizarse soldadura por puntos.
I/O ENBL	I/O ENBL indica que se activan las señales de E/S.
WELD ENBL	Cuando se ilumina, indica que está activada la soldadura con arco.
ARC ESTAB	ARC ESTAB LED se ilumina cuando la soldadura al arco está siendo utilizada.
DRY RUN	Cuando DRY RUN se ilumina indica que se selecciona el modo de operación utilizando la puesta en marcha en seco.
JOINT	JOINT LED se ilumina cuando se selecciona el movimiento JOINT como sistema de coordenadas del movimiento manual
XYZ	XYZ LED se ilumina cuando se selecciona el movimiento cartesiano como sistema de coordenadas del movimiento manual.
TOOL	TOOL LED se ilumina cuando se selecciona el movimiento de la herramienta como sistema de coordenadas del movimiento manual.

Fig. 8. Estados de los indicadores "led" de la consola "Teach Pendant".

Los leds que tendremos en cuenta para nuestro proceso de simulación serán los de la herramienta "*handling tool*" ya que los leds que indican estados de algún proceso de soldadura no serán utilizados (los marcados en gris).

En el teclado del Teach Pendant podemos encontrar los botones que utilizaremos para manejar el software de programación del manipulador. Las principales funcionalidades se especifican en la siguiente tabla:

TECLA	FUNCIÓN
	Las teclas de función (F) se usan para seleccionar un menú de función en la última línea de la pantalla.
	La tecla NEXT se utiliza para habilitar más teclas de función en la página siguiente.
	La tecla MENUS para visualizar el menú de pantalla. La tecla FCTN para visualizar el menú de función.
	La tecla SELECT para visualizar la pantalla de selección del programa. La tecla EDIT para visualizar la pantalla de edición del programa. La tecla DATA para visualizar la pantalla de datos del programa.
	La tecla MAN FCTNS visualiza la pantalla de operación manual.
	La tecla STATUS visualiza la pantalla de posición actual. La tecla I/O visualiza la pantalla de E/S. La tecla POSN visualiza la pantalla de posición actual
	Las teclas SHIFT se utilizan para la habilitación de movimiento del robot, programar los datos de posición, y arrancar un programa.
	Las teclas de movimiento son efectivas mientras se mantiene pulsada una tecla Shift. Se utilizan para la habilitación de movimiento.
	La tecla COORD selecciona un sistema de coordenadas de movimiento manual. Cada vez que se pulsa la tecla COORD, selecciona el siguiente tipo de movimiento en el orden: JOINT, JGFRM, World frame, TOOL, USER. Cuando se pulsa esta tecla mientras se mantiene pulsada una tecla Shift, aparece un menú de movimiento para el cambio del sistema de coordenadas.
	La tecla variación de velocidad . Cada vez que se pulsa varía en el orden: VFINE, FINE, 1%, 5%, 50%, 100%. (Cambio del 5% de la cantidad para el 5% o menos y cambio del 5% de la cantidad para el 5% o más.
	La tecla FWD o la tecla BWD (+tecla SHIFT) arranca un programa. Cuando se libera la tecla shift durante la regeneración, se interrumpe el programa.
	La tecla HOLD provoca que se interrumpa un programa.
	La tecla STEP selecciona paso a paso o realiza operación en ciclo continuo.
	La tecla PREV vuelve a almacenar el estado más reciente. En algunos casos, la tecla no puede volver inmediatamente al estado anterior.
	La tecla ENTER introduce, valida y selecciona un número o un menú.


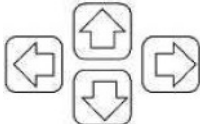

	La tecla BACK SPACE borra el carácter o número inmediatamente anterior al cursor.
	Las teclas de dirección mueven el cursor. El cursor es la zona destacada que puede moverse en la pantalla de la consola de programación. Esta zona llega a ser el objeto de operación (entrada o cambio del valor o contenidos) de la tecla de la consola de programación.
	La tecla ITEM mueve el cursor a una línea cuyo número es especificado.

Fig. 9. Resumen de los botones de la consola "Teach Pendant".

La última parte que podemos distinguir en el "Teach Pendant" es el "display", el cual nos muestra los menús y las distintas pantallas de las que dispone el software de programación del manipulador.

En la siguiente figura se muestra un ejemplo del "display" indicando las principales características en un programa de posicionamiento del brazo:

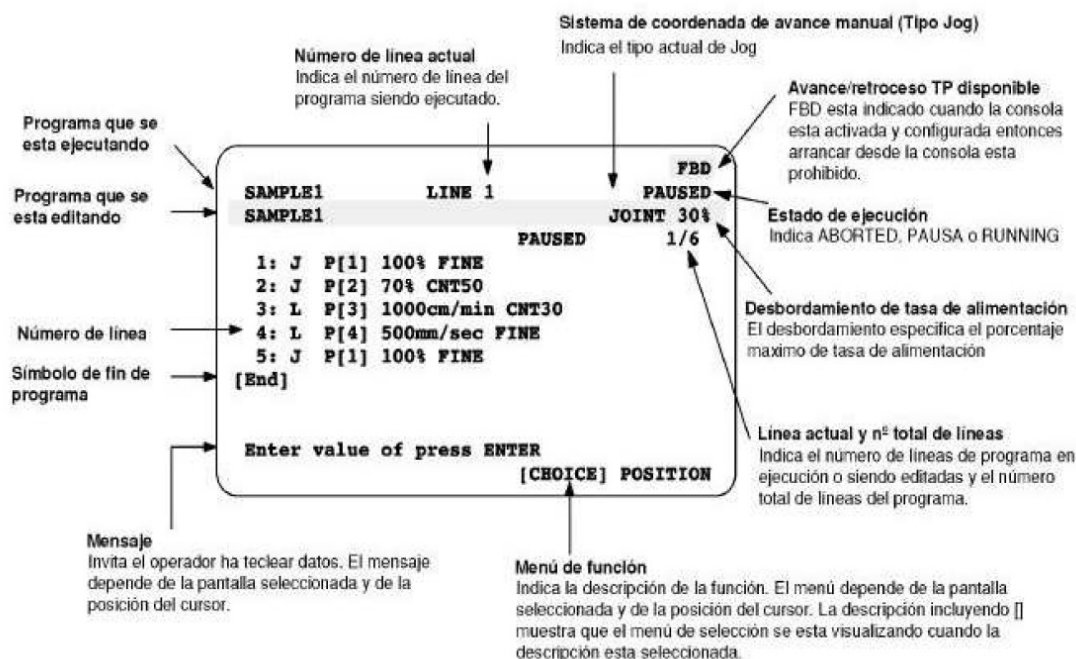


Fig. 10. Ejemplo de programa en "display" TPE.

Conociendo las principales características del dispositivo de programación TPE podemos comenzar a utilizarlo para poder realizar nuestro programa, componiendo así una rutina para el robot y el proceso en general.

En la siguiente figura podemos observar el menú principal de nuestro dispositivo TPE. Vamos a explicar los principales apartados de forma general para después especificar los utilizados en nuestro proceso de programación utilizando el dispositivo TPE de Roboguide:

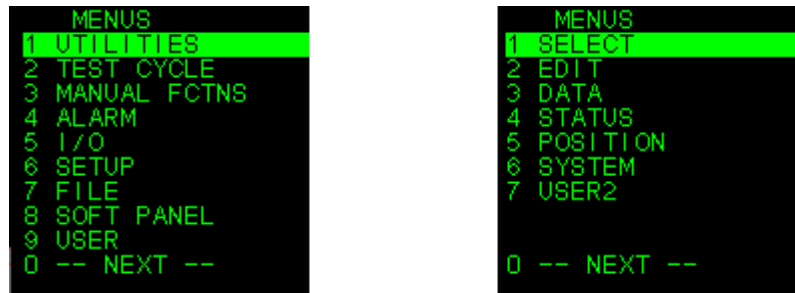


Fig. 11. Menús principales del dispositivo TPE

UTILITIES La pantalla de utilidad se utiliza para visualizar las pistas.

TEST CYCLE La pantalla de ciclo de prueba se utiliza para especificar los datos para la operación de prueba.

MANUAL FCTNS La pantalla de operación manual se utiliza para ejecutar las instrucciones macro.

ALARM La pantalla del historial de alarma muestra la historia y detalles de las alarmas.

I/O La pantalla de E/S para visualizar, forzar, simular y configurar señales de entrada y salida.

SETUP La pantalla de ajuste se utiliza para establecer el sistema.

FILE La pantalla de archivo se utiliza para leer o almacenar archivos.

USER La pantalla del usuario muestra los mensajes del usuario.

SELECT La pantalla de selección del programa se utiliza para enumerar o crear los programas.

EDIT La pantalla de edición del programa se utiliza para corregir y ejecutar el programa.

DATA La pantalla de datos muestra los valores en registros, registros de posición y resto de variables.

STATUS La pantalla de estado muestra el estado del sistema.

POSITION La pantalla de posición actual muestra la posición actual del robot.

SYSTEM La pantalla del sistema se utiliza para establecer las variables del sistema y el "mastering".

Creación de un programa

- 1) TP posicionado en ON
- 2) SELECT
- 3) F2:CREATE
- 4) Definir "Program Name"
(eligiendo en el menú el modo de escritura además de los dígitos del panel).

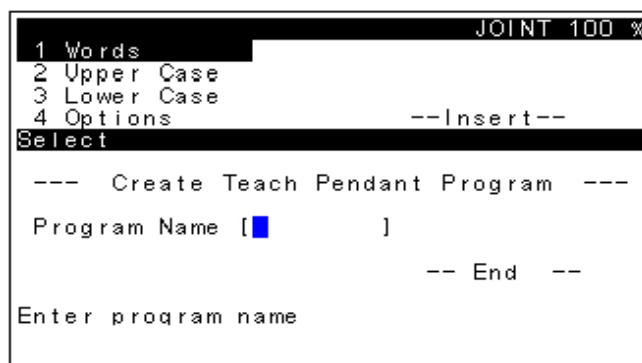


Fig. 12. Pantalla 1 de la creación de programa.

- 5) F2:DETAIL
- 6) F3:EDIT
- 7) F1:END

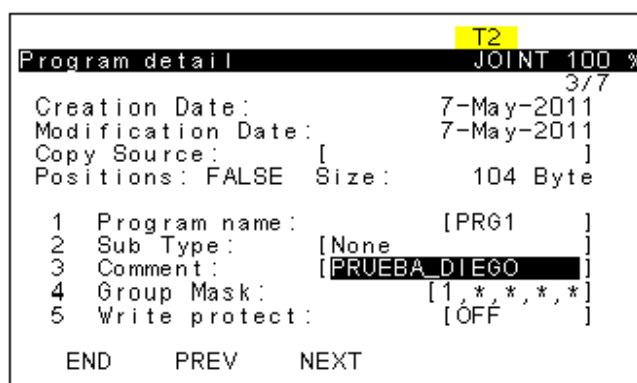


Fig. 13. Pantalla 2 de la creación de programa.

Definición de un punto

- 1) F1:POINT
- 2) Selección del punto a modificar.
- 3) F1:ED_DEF
- 4) Introducción de coordenadas y resto de parámetros.

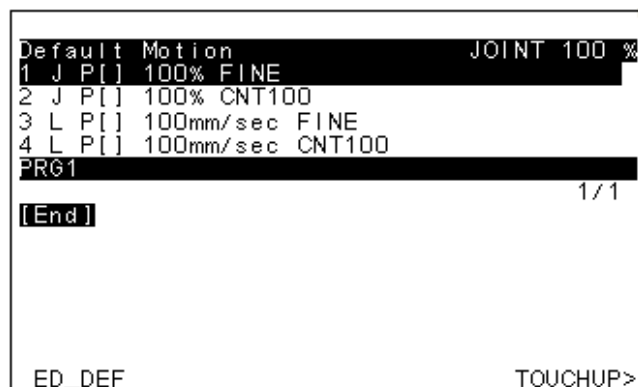


Fig. 14. Pantalla 1 de la definición de un punto.

En la siguiente figura se especifica cada uno de los parámetros a configurar en la definición de un punto en un programa para una rutina:

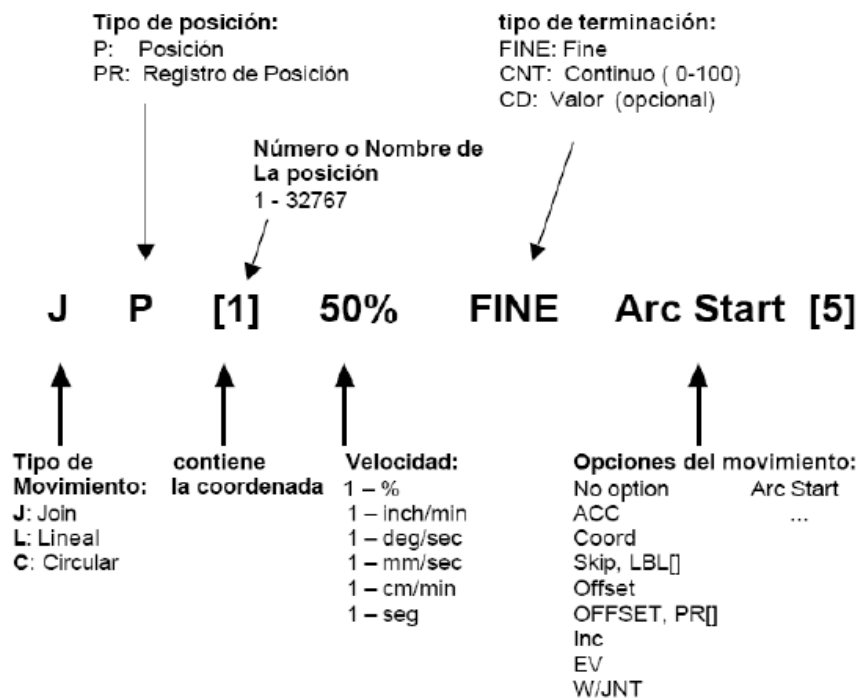


Fig. 15. Estructura para la definición de un punto en la controladora robótica.

Centro de herramienta (TCP):

Cada vez que grabamos un punto éste se nos puede representar en grados y en coordenadas cartesianas. En coordenadas cartesianas, las cotas grabadas, son las del TCP (*Tool Center Point* o Punto Central de la Herramienta), respecto del origen del sistema de coordenadas cartesianas activo en ese momento y elegido previamente por el usuario. (WORLD por defecto). Por defecto el TCP se encuentra en el centro de la placa del eje más externo del robot.

El TCP es el origen de la referencia herramienta. Cuando se crea una referencia de herramienta, el TCP se desplaza al extremo de la herramienta utilizada. La referencia herramienta puede ser orientada según el eje de ataque de esa herramienta.

Distinguimos 2 tipos de herramientas:

Herramienta simple:

Una herramienta simple es una herramienta en la cual el eje de ataque es paralelo al eje Z de la herramienta por defecto.

En este caso la orientación de la herramienta no cambia respecto a la herramienta por defecto; solo se desplaza el TCP. El método de aprendizaje de los 3 puntos es el que se elige para memorizar la herramienta.

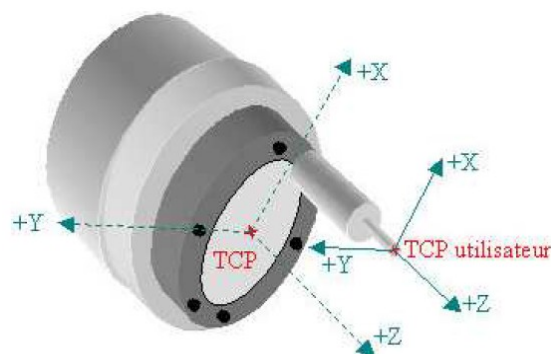


Fig. 16. Herramienta simple en TCP.

Herramienta compleja:

Una herramienta compleja es una herramienta en la cual el eje de ataque no es paralelo al eje Z de la herramienta por defecto.

En este caso el TCP está desplazado y su orientación está redefinida.

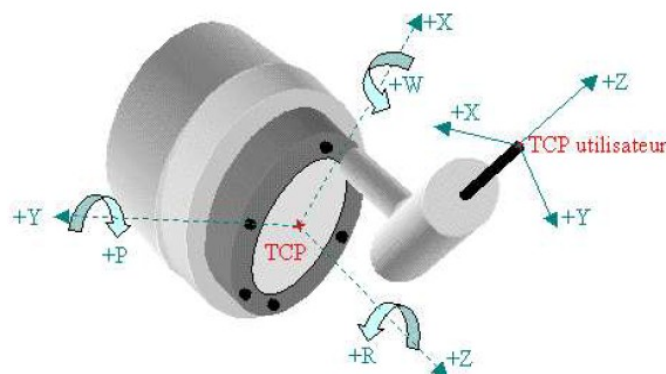


Fig. 17. Herramienta compleja en TCP.

Para definir una herramienta seleccionar:

- 1) MENU
- 2) SETUP
- 3) F1: [TYPE]
- 4) FRAMES
- 5) F3: [OTHER]
- 6) TOOL
- 7) ENTER

Y accederemos a la pantalla que se muestra a continuación:

PRG1	LINE 0	AUTO	ABORTED
SETUP Frames		JOINT 100 %	
Tool Frame Setup/ Direct Entry			1/10
	X	Y	Z
1:	0.0	0.0	436.0
2:	0.0	0.0	0.0
3:	0.0	0.0	0.0
4:	0.0	0.0	0.0
5:	0.0	0.0	0.0
6:	0.0	0.0	0.0
7:	0.0	0.0	0.0
8:	0.0	0.0	0.0
9:	0.0	0.0	0.0
Active TOOL \$MNUTOLNUM[1] = 1			
I TYPE I	DETAIL	IOTHER I	CLEAR SETIND

Fig. 18. Pantalla 1 de la definición de herramienta.

Podemos especificar el método de aprendizaje presionando el la pantalla anterior F2:DETAIL y posteriormente F2:METHOD, accediendo a la figura que sigue a continuación:

PRG1	LINE 0	AUTO	ABORTED
SETUP Frames		JOINT 100 %	
Tool Frame Setup/ Direct Entry			1/7
Frame Number: 1			
1 Comment:			PINZA
2 X:		0.000	
3 Y:		0.000	
4 Z:		436.033	
5 W:		0.000	
6		0.000	
7		0.000	
1 Three Point			
2 Six Point			
3 Direct Entry			
Act			UM[1] = 1
I TYPE I	I METHOD I	FRAME	

Fig. 19. Pantalla 2 de la definición de herramienta.

1) Tres puntos:

El objeto de este método es el de desplazar el TCP al extremo de la herramienta utilizada. Para ello tenemos que marcar un mismo punto con 3 orientaciones diferentes y memorizar esas posiciones.

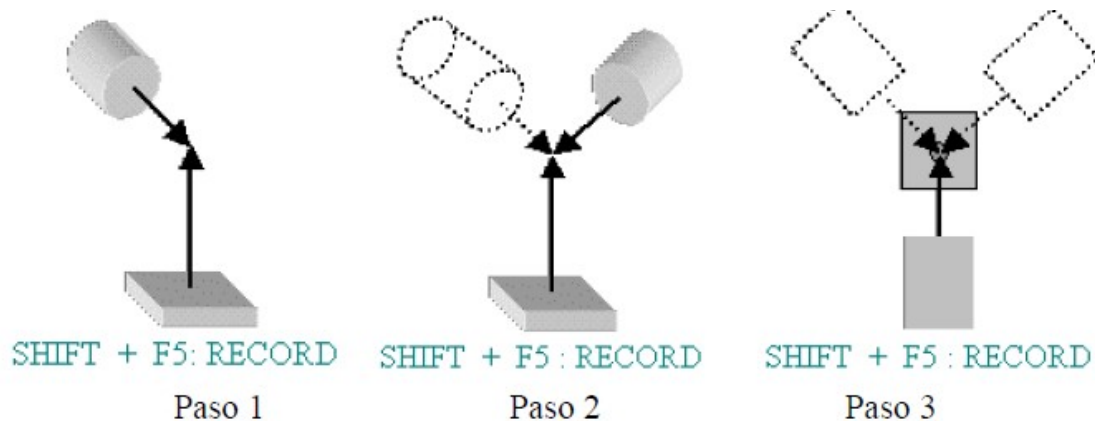


Fig. 20. Definición de punto de origen de la herramienta.

Cuando los tres puntos se han memorizado, las coordenadas x, y, z del nuevo TCP, son visualizadas en la parte superior de la ventana. Estas coordenadas son dadas respecto al TCP original de fábrica.

El sentido de la coordenada Z del TCP creado por el método 3P es el mismo que la del TCP original del robot.

Para utilizar este método de definición de herramienta en Roboguide encontraremos la siguiente pantalla:

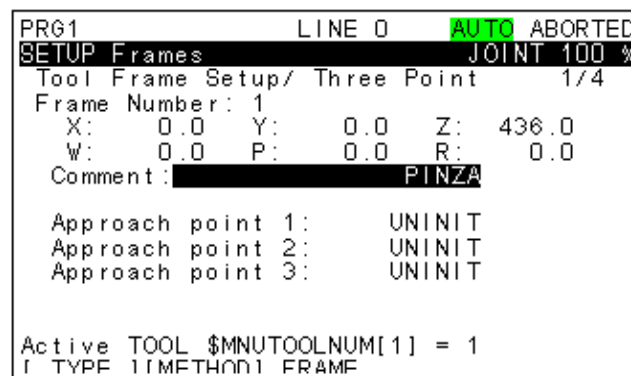


Fig. 21. Pantalla de definición de origen de herramienta por el método de 3 puntos.

2) Seis puntos

El objeto de este método es el de desplazar el TCP original del robot a un punto concreto de la herramienta utilizada y de reorientar la herramienta en base a ese punto.

El sentido de la coordenada Z del TCP creado por el método 6P es diferente que la del TCP original del robot. En este caso es impuesta por el usuario.

Pasos 1, 2 y 3:

Los tres primeros pasos son idénticos a los tres primeros pasos que el método de los tres puntos. El TCP está definido y ahora debemos reorientar la herramienta y memorizar tres puntos adicionales.

Paso 4: Orient Origine Point

Para memorizar el punto de origen de la orientación, el eje OZ de la herramienta debe estar colocado verticalmente, como en la figura siguiente:

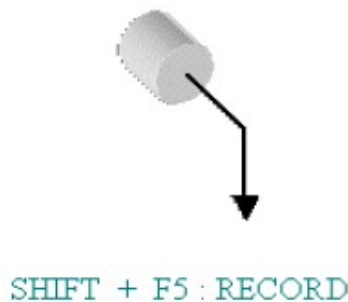


Fig. 22. Definición de Orient Origine Point de la herramienta.

Paso 5: X Direction Point

Definiremos ahora la orientación y el sentido del eje X. Para este paso y el siguiente, es más práctico moverse en el sistema WORLD, con el fin de asegurar que desplazamos horizontalmente el eje OZ de la herramienta. WORLD +/- X y/o +/- Y

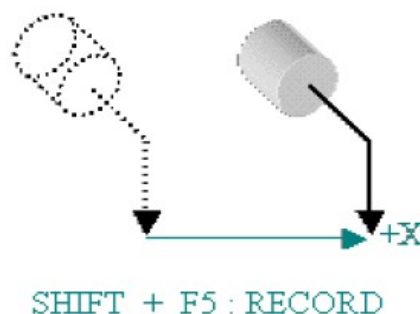


Fig. 23. Definición de X Direction Point de la herramienta.

Paso 6: Z Direction Point

Para dar la dirección en Z, es preciso re-posicionarse sobre el punto de origen de la orientación. Para ello colocar el cursor sobre la línea « Orient Origine Point » y después pulsar SHIFT + F4: MOVE_TO.

El robot se re-posicionará sobre el punto memorizado en el paso 4.

Para definir la dirección y el sentido del eje Z, como muestra la imagen:

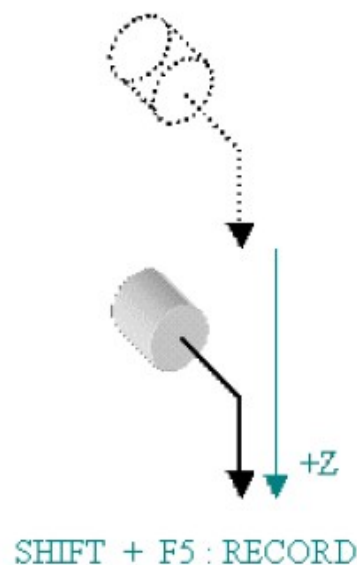


Fig. 24. Definición de Z Direction Point de la herramienta.

Cuando los seis puntos están memorizados, las coordenadas x, y, z del nuevo TCP y las orientaciones w, p y r de la nueva herramienta son visualizadas en la parte superior de la ventana.

PRG1	LINE 0	AUTO	ABORTED
SETUP Frames		JOINT 100 %	
Tool Frame Setup/ Six Point		1/7	
Frame Number: 1			
X: 0.0	Y: 0.0	Z: 436.0	
W: 0.0	P: 0.0	R: 0.0	
Comment:	PINZA		
Approach point 1:	UNINIT		
Approach point 2:	UNINIT		
Approach point 3:	UNINIT		
Orient Origin Point:	UNINIT		
X Direction Point:	UNINIT		
Z Direction Point:	UNINIT		
Active TOOL \$MNUTOLNUM[1] = 1			
[TYPE] [METHOD] FRAME			

Fig. 25. Pantalla de definición de origen de herramienta por el método de 6 puntos.

3) Entrada directa

En este método, las coordenadas y orientación de la herramienta a definir deben ser perfectamente conocidos.

Estas coordenadas serán introducidas directamente a mano en la ventana siguiente:

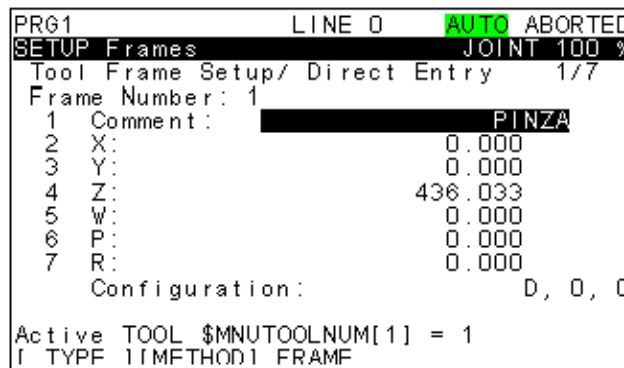


Fig. 26. Pantalla de definición de origen de herramienta por el método entrada directa.

Sistema de referencia definido por usuario

Un sistema de referencia de usuario USER FRAME es un sistema de referencia tridimensional y cartesiano sobre el cual se memorizan todas las posiciones de un de terminado programa TP. El TCP se mueve y reorienta en base a ese sistema siempre que movamos el robot en modo USER.

Si no hay definido ningún sistema de referencia usuario, por defecto, las posiciones se referirán al sistema de coordenadas WORLD.

La figura que se muestra a continuación representa un sistema de referencia personalizado consiguiendo un entorno de trabajo modificado por el usuario:

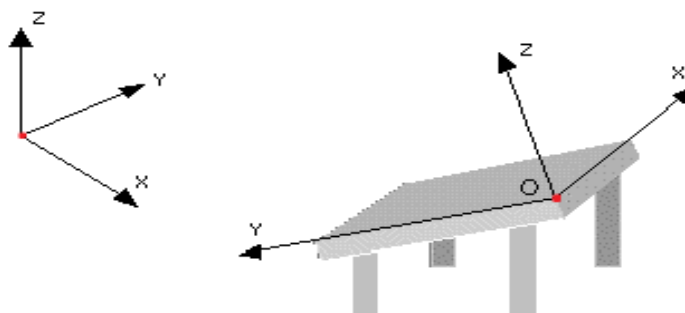


Fig. 27. Sistema de referencia definido por usuario (UFRAME).

Métodos de configuración:

Para poder definir un sistema de referencia usuario podemos hacerlo mediante utilizando las siguientes instrucciones:

- 1) MENU
- 2) SETUP
- 3) F1:TYPE
- 4) FRAMES
- 5) F3:OTHER
- 6) USER
- 7) ENTER

PRG1	LINE 0	AUTO	ABORTED
SETUP Frames		JOINT 100 %	
User Frame Setup/ Direct Entry			1/9
X Y Z			Comment
1:	0.0	0.0	0.0 UFrame1
2:	0.0	0.0	0.0 UFrame2
3:	0.0	0.0	0.0 UFrame3
4:	0.0	0.0	0.0 UFrame4
5:	0.0	0.0	0.0 UFrame5
6:	0.0	0.0	0.0 UFrame6
7:	0.0	0.0	0.0 UFrame7
8:	0.0	0.0	0.0 UFrame8
9:	0.0	0.0	0.0 UFrame9
Active UFRAME \$MNUFRAMENUM[1] = 0			
[TYPE] DETAIL [OTHER] CLEAR SETIND >			

Fig. 28. Pantalla 1 de los métodos de configuración de UFRAME.

Accederemos a la pantalla anterior y podremos configurar con el cursor la herramienta determinada.

- 8) F2:DETAIL
- 9) F2:METHOD

PRG1	LINE 0	AUTO	ABORTED
SETUP Frames		JOINT 100 %	
User Frame Setup/ Direct Entry			1/7
Frame Number: 1			
1 Comment:			UFrame1
2 X:			0.000
3 Y:			0.000
4 Z:			0.000
5 W:			0.000
6 P:			0.000
7 R:			0.000
Configuration: D, 0, 0			
Active UFRAME \$MNUFRAMENUM[1] = 0			
[TYPE] [METHOD] FRAME MOVE_TO RECORD			

Fig. 29. Pantalla 2 de los métodos de configuración de UFRAME.

Elegiremos ahora uno de los tres métodos de aprendizaje para definir la referencia de usuario. El origen del sistema de referencia se desplazará al sitio deseado y la posición y orientación siguiendo las tres direcciones que se definan:

3 Puntos:

Seleccionaremos en el menú:

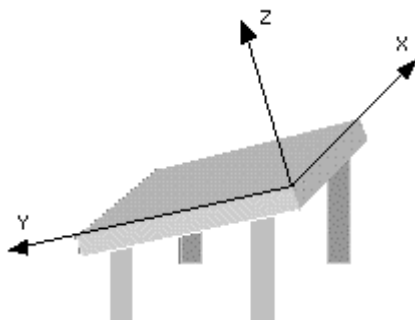
- 1) F2:METHOD
- 2) THREE POINT

Con este método definiremos dos rectas que se cruzan determinan un plano, con origen fijo en el punto de cruce y Z perpendicular al plano definido.

Orient Origine Point

Memorizaremos el punto de origen para determinar la referencia personalizada por usuario.

Con el cursor nos desplazaremos y determinaremos cada uno de los valores a definir.



PRG1	LINE 0	AUTO	ABORTED
SETUP Frames			JOINT 100 %
User Frame Setup/ Three Point			1/4
Frame Number: 1			
X:	0.0	Y:	0.0
W:	0.0	P:	0.0
Z:		R:	0.0
Comment:	UFrame1		
Orient Origin Point:	UNINIT		
X Direction Point:	UNINIT		
Y Direction Point:	UNINIT		
Active UFRAME \$MNUFRAMENUM[1] = 0			
[TYPE][METHOD] FRAME			

Fig. 30. Pantalla de definición de UFRAME con el método de los 3 puntos.

Instrucciones de programación de rutinas con registros y registros de posición.

Las variables disponibles a utilizar son:

Los registros: real (32 bits) o entero

Los registros de posición: puntos en coordenadas joint, puntos en coordenadas cartesianas o matrices.

Estas son variables globales (todos los programas tienen acceso a todos los registros y registros de posición).

Los registros:

Hay un máximo de 256 (configurables) y permiten ser comentados con un nombre. Los registros pueden definirse de las siguientes formas:

Directa:

$R[1] = 2$ -> el valor es guardado directamente en $R[1]$

Indirecta:

$R[R[1]] = 5$ -> el registro afectado depende del valor contenido en $R[1]$

Si $R[1] = n$, por tanto el valor 5 es guardado en $R[n]$.

- En un registro es posible almacenar el resultado de una operación aritmética.

$R[n] = [\text{valor}] [\text{operador}] [\text{valor}]$

El [operador] puede ser:

- suma (+)
- resta (-)
- multiplicación (*)
- división (/)
- división entera (DIV)
- resto (MOD)

El [valor] puede ser:

- constante
- entrada-salida analógico $AI[n]/AO[n]$
- entrada-salida digital $DI[n]/DO[n]$
- entrada-salida grupo $GI[n]/GO[n]$
- entrada-salida de robot $RI[n]/RO[n]$
- registro $R[n]$
- elemento de un registro de posición $PR[i,j]$

Para insertar en un programa -> $F1:[INST]$ -> Registers.

Para visualizar la lista de registros y su contenido -> $DATA$ -> $F1:[TYPE]$ -> Registers.

Los registros de posición:

Un registro de posición almacena un punto.

Los registros de posición pueden definirse de las siguientes formas:

Directa:

$PR[1] = P[1]$ -> el punto es guardado directamente en $PR[1]$

Indirecta:

$PR[R[1]] = P[3]$ -> el registro de posición afectado depende del valor contenido en $R[1]$

Si $R[1] = n$, entonces el punto $P[3]$ está almacenado en $PR[n]$.

En un registro de posición es posible almacenar un punto o una operación de punto.

$PR[n] = [\text{punto}] [\text{operador}] [\text{punto}]$

El [operador] puede ser:

- suma (+)
- resta (-)

El [punto] puede ser:

- posición $P[n]$
- registro de posición $PR[n]$
- posición actual del robot en grados eje por eje JPOS
- posición actual del robot en cartesianas LPOS

Los registros de posición son también accesibles elemento por elemento.

Por ejemplo, la coordenada j de $PR[i]$ está definida por $PR[i,j]$

$PR[1,2] = 250$ -> la coordenada Y de $PR[1]$ está inicializada a 250mm.

O indirectamente

$R[1] = 1$

$R[2] = 2$

$PR[R[1],R[2]] = 250$ -> la coordenada Y de $PR[1]$ está inicializada a 250mm.

Cada posición y orientación es por tanto accesible independientemente.

Para insertar en un programa -> 1:[INST] -> registers

Para visualizar la lista de registros y su contenido -> DATA -> F1:[TYPE] -> Position Registers.

Instrucciones de salto condicional:

Una instrucción de salto condicional permite efectuar un salto (o bucle) a una etiqueta situada en el mismo programa si (y sólo si) ciertas condiciones son verdaderas.
F1:[INST] ->F/SELECT.

Instrucción IF

Realiza un salto en función de una condición verdadera:

IF [valor1] [operador] [valor2] [salto]

El [valor1] puede ser:

- registro R[n]
- entradas-salidas analógicas AI[n]/AO[n]
- entradas-salidas digitales DI[n]/DO[n]
- entradas-salidas de grupo GI[n]/GO[n]
- entradas-salidas de robot RI[n]/RO[n]

El [operador] puede ser:

- comparación (=)
- diferenciación (< >)
- menor (<)
- mayor (>)
- menor o igual (<=)
- mayor o igual (=>)

El [valor2] puede ser:

- constante
- ON
- OFF
- registro R[n]
- entradas-salidas analógicas AI[n]/AO[n]
- entradas-salidas digitales DI[n]/DO[n]
- entradas-salidas de grupo GI[n]/GO[n]
- entradas-salidas de robot RI[n]/RO[n]

El [salto] puede ser:

- JMP LBL[n]
- CALL programa

Instrucción SELECT: Efectúa uno o varios saltos en función del valor de un registro.

SELECT R[n] = [valor1], [salto]

[valor2], [salto]

[valor n], [salto]

ELSE, [salto]

Los [valores] pueden ser:

- constante
- registro R[n]

Los [saltos] pueden ser:

- JMP LBL[n]
- CALL programa

No olvidar la instrucción ELSE, ya que tiene en cuenta el resto de los valores posibles del registro R[n].

Instrucciones de espera:

Las instrucciones de espera retardan la ejecución de un programa mediante un tiempo especificado o hasta que una condición sea verdadera.

F1:[INST] -> WAIT.

Temporización: Retarda la ejecución de un programa durante un tiempo especificado. La duración se expresa en segundos; hay un mínimo de 0,01 segundos

WAIT [tiempo].

El [tiempo] puede ser:

- constante
- registro R[n].

Espera de una condición: Retarda la ejecución de un programa hasta que la condición se cumpla.

WAIT [valor1] [operador] [valor2] [tiempo].

El [valor1] puede ser:

- registro R[n]
- entradas-salidas digitales DI[n]/DO[n]
- entradas-salidas de robot RI[n]/RO[n]

El [operador] puede ser:

- comparación (=)
- diferenciación (<>)

- El [valor2] puede ser:

- constante
- ON
- OFF
- registro R[n]
- entradas-salidas digitales DI[n]/DO[n]
- entradas-salidas de robot RI[n]/RO[n]

- El [tiempo] puede ser:
- FOREVER -> espera mientras la condición no se cumpla
- TIMEOUT LBL[n] -> espera el tiempo especificado en la variable *timeout* (\$WAITTMOUT), después salta a "label n" si la condición no se ha cumplido.

Registros periféricos de I/O (OUP):

Registros de entrada (UI)

*IMSTP Siempre activa	<p>*IMSTP es la señal de parada inmediata del software. *IMSTP es una señal normal de OFF que se mantiene en ON. Cuando se pone a OFF sucede que:</p> <ul style="list-style-type: none"> - Se detiene el programa si se estaba ejecutando. - Se detiene inmediatamente el robot y actúan los frenos. - Se desconecta la tensión de los servos <p>Cuando se pierde esta señal, se visualiza el código de error SRVO-037 *IMSTP input. Esta señal esta siempre activa</p>
*HOLD Siempre activa	<p>*HOLD es la señal de detención externa. *HOLD es una señal de normal de OFF, que se mantiene en ON. Cuando se pone en OFF, sucede lo siguiente:</p> <ul style="list-style-type: none"> - Detiene le programa en ejecución. - Desciende la velocidad a una controlada y se detiene. - El freno en detención desconecta la tensión de los servos al parar el robot.
*SFSPD Siempre activa	<p>*SFSPD es la señal de entrada de velocidad de seguridad. Esta señal se conecta generalmente a la valla de seguridad. *SFSPD es una señal normal de OFF, que se mantiene en ON. Cuando pasa a OFF, sucede lo siguiente:</p> <ul style="list-style-type: none"> - Detiene el programa en ejecución. - Reduce la velocidad a un valor definido en una variable del sistema. Este valor no puede incrementarse mientras *SFSPD este en OFF. - Visualiza el mensaje de error SYST009 - No permite una condición de marcha REMOTE. Las entradas de marcha del UOP o del SOP se inhabilitan cuando SFSPD se pone en OFF y tan solo el terminal de enseñanza tiene control sobre el movimiento con la velocidad limitada.
FAULT_RESET Siempre activa	<p>FAULT_RESET es la señal externa de aceptar el fallo. Cuando se recibe esta señal, sucede lo siguiente:</p> <ul style="list-style-type: none"> - Cesa el estado de error. - Los servos reciben tensión. - El programa detenido no continuará.
START Activa cuando el robot se halla en condición de mando remoto (CMDENBL=ON)	<p>START es la entrada de marcha remita. La función de esta señal, depende de la variable del sistema \$SHELL_CFG.\$CON_ONLY se fija a FALSE, la señal de START:</p> <ul style="list-style-type: none"> - Hace seguir a un programa detenido. - Si el programa se aborta, el programa seleccionado en este momento empieza en la posición del cursor.

Fig. 31. Resumen de registros periféricos de entrada.

Estas señales permiten manejar el robot a distancia por medio de un panel de operador (UOP) o PLC. Las funciones de las salidas UOP (UI[n] UO[n]) están predefinidas y pueden ser cableadas sobre cartas modulares digitales o configuradas mediante cartas de bus de campo (Interbus, Profibus, Devicenet,...) 18 entradas y 20/24 salidas (4 opcionales) pueden ser conectadas (mínimo 8 entradas o salidas).

Registros de salida (UO)

CMDENBL	CMDENBL es la salida de habilitación de órdenes. Esta salida indica que el robot se halla en una situación de mando remoto. Esta señal se pone en ON cuando en interruptor remoto se conmuta a ON. Esta salida solamente permanece activa cuando el robot no se halla en situación de fallo. Cuando SYSRO esta en OFF, CMDENBL está en OFF. Esta señal se pone activa cuando se cumplen todas las condiciones siguientes: <ul style="list-style-type: none"> - Terminal de enseñanza inhabilitado - Interruptor remoto en ON - Entrada SFPD en ON - Entrada ENBL en ON - Variable del sistema \$RMT_MASTER a 0 - No se halla en modo paso a paso (STEP) - Interruptor de selección en modo Auto.
SYSRDY	SYSRDY es la salida de sistema preparado. Esta salida indica que los servos se hallan activos.
PROGRUN	PROGRUN es la salida de programa funcionando. Esta salida se pone activa cuando hay un programa funcionando.
PAUSED	PAUSED es la salida de programa detenido. Esta salida se pone activa cuando se ha detenido un programa.
HELD	HELD es la salida de detención. Esta salida se pone activa cuando la entrada *HOLD del UOP se halla en OFF o se ha accionado el pulsador HOLD de la TP
FAULT	FAULT es la salida de error. La salida se pone activa cuando un programa entra en situación de error.

Fig. 32. Resumen de registros periféricos de salida.

Para llegar a la pantalla de configuración de las entradas y salidas del robot deberemos pulsar:

- 1) MENU
- 2) 5 I/O
- 3) pulsando F1:[TYPE] vemos los diferentes tipos de entradas y salidas.

PRG1	LINE 0	AUTO	ABORTED
I/O Cell Outputs		JOINT 100 %	1/24
1 Cell Intface	TYPE #	SIM	STATUS
2 Digital	DO[0]	U	***
3 Analog	DO[0]	U	***
4 Group	DO[0]	U	***
5 Robot	DO[0]	U	***
6 UOP	DO[0]	U	***
7 SOP	DO[0]	U	***
8 Interconnect	DO[0]	U	***
9 Link Device	DO[0]	U	***
0 -- NEXT --	DO[0]	U	***
+-----+			
I TYPE I	CONFIG	IN/OUT	SIM UNSIM >

Fig. 33. Pantalla de configuración de registros periféricos I/O.

En cualquier pantalla de entradas y salidas, si pulsamos F2 [CONFIG] entramos dentro de la pantalla de configuración de las mismas.

UOP:

Los parámetros de estas I/O son iguales que los de las I/O digitales; hay que tener en cuenta que los rangos de estos grupos no se pueden superponer en el mapa de I/O.

Las UOP marcadas con un asterisco (*) funcionan con lógica negada.

Usando la tecla F3 [IN/OUT] alternamos la pantalla de configuración de entradas con la de salidas y viceversa, mostrando los menús de las siguientes figuras:

PRG1	LINE 0	AUTO	ABORTED
I/O Cell Inputs		JOINT 100 %	
		1/15	
INPUT SIGNAL	TYPE #	SIM	STATUS
1 Tryout Mode	DI[0]	U	***
2 User in 1:	DI[0]	U	***
3 User in 2:	DI[0]	U	***
4 User in 3:	DI[0]	U	***
5 User in 4:	DI[0]	U	***
6 User in 5:	DI[0]	U	***
7 User in 6:	DI[0]	U	***
8 User in 7:	DI[0]	U	***
9 User in 8:	DI[0]	U	***
10 User in 9:	DI[0]	U	***
11 User in 10:	DI[0]	U	***
12 User GIN 1:	GI[0]	U	*****
13 User GIN 2:	GI[0]	U	*****
14 User GIN 3:	GI[0]	U	*****
15 User GIN 4:	GI[0]	U	*****
[TYPE]	CONFIG	IN/OUT	SIM UNSIM >
PRG1	LINE 0	AUTO	ABORTED
I/O Cell Outputs		JOINT 100 %	
		1/24	
OUTPUT SIGNAL	TYPE #	SIM	STATUS
1 Input Simulated	DO[0]	U	***
2 OVERRIDE = 100	DO[0]	U	***
3 Tryout Status	DO[0]	U	***
4 Heartbeat	DO[0]	U	***
5 MH Fault	DO[0]	U	***
6 MH Alert	DO[0]	U	***
7 Refpos1[1]	DO[0]	U	***
8 Refpos1[2]	DO[0]	U	***
9 Refpos1[3]	DO[0]	U	***
10 Robot motion G1	DO[0]	U	***
11 User out 1:	DO[0]	U	***
12 User out 2:	DO[0]	U	***
13 User out 3:	DO[0]	U	***
14 User out 4:	DO[0]	U	***
15 User out 5:	DO[0]	U	***
16 User out 6:	DO[0]	U	***
17 User out 7:	DO[0]	U	***
18 User out 8:	DO[0]	U	***
19 User out 9:	DO[0]	U	***
20 User out 10:	DO[0]	U	***
21 User GOUT 1:	GO[0]	U	*****
22 User GOUT 2:	GO[0]	U	*****
23 User GOUT 3:	GO[0]	U	*****
24 User GOUT 4:	GO[0]	U	*****
[TYPE]	CONFIG	IN/OUT	SIM UNSIM >

Fig. 34. Estado de configuración de registros periféricos de I/O.

Configuración del Robot.

Pantalla de configuración del sistema:

Para obtener un resumen de la configuración del robot accederemos de la siguiente forma a través del controlador TPE.

```

PRG1                LINE 0    AUTO ABORTED
System/Config       JOINT 100 %
1/44
1 Use HOT START:    FALSE
2 I/O power fail recovery: RECOVER ALL
3 Autoexec program [*****]
  for Cold start:
4 Autoexec program [*****]
  for Hot start:
5 HOT START done signal: DO[ 0]
6 Restore selected program: TRUE
7 Enable UI signals: TRUE
8 START for CONTINUE only: FALSE
9 CSTOP1 for ABORT: FALSE
10 Abort all programs by CSTOP1: FALSE
11 PROD_START depend on PNSTROBE: FALSE
12 Detect FAULT_RESET signal: FALL
13 Use PPABN signal: < *GROUPS * >
14 WAIT timeout: 30.00 sec
15 RECEIVE timeout: 30.00 sec
16 Return to top of program: TRUE
17 Original program name(F1): [PRG ]
18 Original program name(F2): [MAIN ]
19 Original program name(F3): [SUB ]
20 Original program name(F4): [TEST ]
21 Original program name(F5): [*****]
22 Default logical command: < *DETAIL * >
23 Maximum of ACC instruction: 150
24 Minimum of ACC instruction: 0
25 WJNT for default motion: *****
26 Auto display of alarm menu: FALSE
27 Force Message: ENABLE
28 Reset CHAIN FAILURE detection: FALSE
29 Allow Force I/O in AUTO mode: TRUE
30 Allow chg. ovrd. in AUTO mode: TRUE
31 Signal to set in AUTO mode DO[ 0]
32 Signal to set in T1 mode DO[ 0]
33 Signal to set in T2 mode DO[ 0]
34 Signal to set if E-STOP DO[ 0]
35 Set if INPUT SIMULATED DO[ 0]
36 Sim. Input Wait Delay: 0.00 sec
37 Set if Sim. Skip Enabled DO[ 0]
38 Set when prompt displayed DO[ 0]
39 Signal if OVERRIDE = 100 DO[ 0]
40 Hand broken : < *GROUPS * >
41 Remote/Local setup: OP panel key
42 External I/O(ON:Remote):DI [ 0]
43 UOP auto assignment: None
44 Multi Program Selection: TRUE

```

Fig. 35. Resumen de la configuración del robot.

Configuración de la red profibús.

Accedemos a la pantalla de configuración de la red *profibús* mediante las teclas:

<Menu> 6 Setup

F1 [TYPE] 0 NEXT 7 PROFIBÚS

F3 [OTHER]

1- Slave- Configuración del profibús del esclavo

2- Master- Configuración del profibús del master.

3- Bus Param.- Configuración de los parámetros del profibús.

4- Slave Param.- Pantalla de configuración de los parámetros de los diferentes esclavos del robot.

Dentro de la pantalla "*Slave Param*". Pulsando F2 [DETAIL] habilitamos/deshabilitamos el esclavo, configuramos su dirección, establecemos el nombre, bytes de entrada y salida, flags...

Configuración de las entradas y salidas

I/O digitales:

Rack- Dirección de la tarjeta de I/O del robot (66 Master; 67 Slave

Slot- Lugar físico donde se encuentra situada la tarjeta de I/O.

Range- Número de I/O "asociadas".

Start- Número de bit por el que empieza el rango de I/O.

Grupos:

Rack- Dirección de la tarjeta de I/O del robot (66 Master; 67 Slave

Slot- Lugar físico donde se encuentra situada la tarjeta de I/O.

Range- Número de I/O "asociadas".

Start- Número de bit por el que empieza el rango de I/O.

Configuración del Payload:

Para configurar un Payload tendremos que acceder a la pantalla de sistema:

<Menú> 0 NEXT 6 SYSTEM

F1 [TYPE] 6 MOTION

Configuramos el peso, la posición del centro de gravedad y le asignamos un nombre. Tenemos hasta un máximo de 10 payloads diferentes.

Profibús: sistema de transmisión de información que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción.

Payload: es el peso de la "muñeca robótica" incluyendo el peso de la EOAT (End of Arm Tool) y la pieza. El "payload" es característico de cada modelo de robot y nos permite elegir los robots aptos para los procesos previamente especificados.

Configuración de Macros

Una MACRO es un programa que efectúa una operación específica cuya ejecución puede ser comandada por:

- La activación de una tecla de usuario del Teach Pendant (UK[n]). Group Mask (*;*;*;*;*)
- La activación de una tecla de usuario del Teach Pendant SHIFT + (SU[n])
- La activación de una tecla de usuario del controlador (opción) (SP[n])
- La selección de un ítem del menú MANUAL FCTNS (MF[n])
- Instrucción CALL Teclas de Macro del TPE
- Instrucción RUN
- La activación de una entrada (DI[n]/RI[n]). Para ampliar \$MACROMAXDRI.
- La activación de una entrada UI[n].

Para crear una Macro deben de realizarse los siguientes pasos:

- Pulsar la tecla SELECT.
- Pulsar F2:CREATE.
- Damos un nombre a la macro.
- Pulsamos la tecla F2:DETAIL.
- En el apartado 2 (Subtype), pulsamos F\$:CHOICE y seleccionamos la opción MACRO y pulsamos END.
- Editamos la macro que hemos creado y escribimos el código que queremos que se ejecute cuando llamamos a esta macro, por ejemplo desde un programa.
- Posteriormente configuramos la macro, para eso pulsaremos:
 - 1) MENU
 - 2) 6 SETUP
 - 3) F1:TYPE
 - 4) 3 MACRO
- En la pantalla "Macro Command", colocamos el cursor en *Program*.
- Pulsamos F4:CHOICE y nos aparecerá el listado de programas (escogemos el definido)
- Colocamos el cursor sobre la columna *Assign*, y pulsamos F4:CHOICE.
- Seleccionamos MF, para que la macro pueda ser llamada desde otro programa.

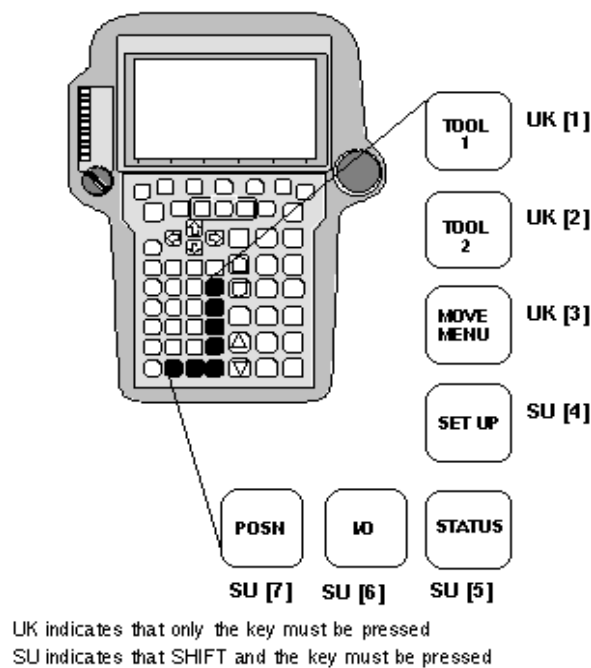


Fig. 36. Botones de ejecución directa de macros.

Ajuste de límite de ejes.

Existen 3 tipos de limitación del recorrido de ejes:

- Límites de software.
- Límites eléctricos.
- Límites mecánicos.

Los límites de software fijos:

Estos son los primeros límites que se encuentra el robot (si están correctamente definidos). Cuando un límite de software es alcanzado, el robot no da fallo, simplemente se para y no permite movimiento en ese sentido. Para poder volver a mover el robot, es suficiente con mover el robot en sentido inverso. Si queremos acceder a la pantalla de la imagen deberemos proceder como se indica a continuación:

- 1) MENU
- 2) 0-NEXT
- 3) 6-SYSTEM
- 4) F1-[TYPE],
- 5) Axis limits.

PRG1	LINE 0	AUTO	ABORTED
SYSTEM	Axis Limits	JOINT 100 %	
AXIS	GROUP	LOWER	UPPER 1/40
1	1	-180.00	180.00 dg
2	1	-45.00	100.00 dg
3	1	-125.00	10.00 dg
4	1	-270.00	270.00 dg
5	0	0.00	0.00 mm
6	0	0.00	0.00 mm
7	0	0.00	0.00 mm
8	0	0.00	0.00 mm
9	0	0.00	0.00 mm
[TYPE]			

Fig. 37. Definición de límites de software fijos.

Para que las modificaciones de límite de software se tengan en cuenta, es preciso apagar y volver a arrancar el controlador.

Límites mecánicos:

Es posible reglar ciertos límites mecánicos dependiendo de los ejes y del modelo del robot.

Si un límite mecánico es alcanzado, se deben verificar los límites eléctricos y los límites de software. Se producirá una alarma de colisión por sobre consumo de motor.

Arranque de programa a distancia vía UI:

Para utilizar las UOP se debe respetar el siguiente protocolo:

- Configurar las señales del sistema UOP.
- Cablear las señales del sistema obligatorias y las que se deseen para control de la instalación.
- Para que la señal de entrada UI [6:START] tenga efecto se han de cumplir dos condiciones:

Habilitar las UI signals:

- 1) MENU
- 2) O-NEXT
- 3) 6-SYSTEM
- 4 F1:TYPE
- 5) 5-CONFIG
- 6) ENABLE UI SIGNALS a TRUE.

El robot nos tiene que dar la señal de salida UO
[1:CMD ENABLE]=ON:

¿Cuándo se produce esta situación?

- 1- UI [1:*IMSTP]=ON, no se recibe ninguna emergencia externa por software.
- 2- UI [2:*HOLD]=ON, no se recibe ningún paro de programa externo.
- 3- UI [3:*SFSPD]=ON, no se recibe ningún paro de programa asociado a un arranque con velocidad predefinida en una variable.
- 4- UI [8:*ENABLE]=ON, se permite la habilitación de movimientos al robot.
- 5- Llave T1, T2, AUTO se encuentra en modo AUTO, con lo que las seguridades externas por hardware quedan habilitadas.
- 6- Controlador en modo REMOTO con lo que se permite el arranque del robot desde un sistema remoto por ejemplo un pulsador de marcha asociado a la entrada UI [6:START] que generará un pulso que tendrá su efecto con el flanco descendente. Para ello R-J2 y R-J3 la Llave LOCAL / REMOTE debe estar posicionada en REMOTE. En el controlador R-J3i deberemos configurar en Menú, 0-Next, 6-System, F1:TYPE, Config, línea 36, Opción,Local/Remote = Remote
- 7- Variable del sistema \$RMT_MASTER=0 si no lo está. MENU, O- NEXT, 6- SYSTEM, F1:TYPE, 2- VARIABLES.
- 8- Asegurarse que la opción "Start For Continue Only" está a false:
MENU, O- NEXT, 6- SYSTEM, F1: TYPE, 5- CONFIG, START FOR CONTINUE ONLY a "FALSE". (si estaba a "True", poner a False y luego OFF/ON para que tome efecto, o modificar la variable \$SHELL_CFG.\$CONT_ONLY=FALSE)
- 9- Teach Pendant en OFF y en condiciones de no STEP (paso a paso).
- 10- UO [2:SYS READY]=ON, el robot no tiene ningún fallo. Reset de fallos: Reset de fallos externos vía software a través de las UI's. Reset de fallos externos vía hardware (Emerg. Externas, Fence Open,...)
- 11- FCTN, 1-Abort All, Select, seleccionar el programa arrancar.
- 12- UI[6:Start] tiene su efecto sobre el robot con flanco descendente.

4.2.2 Editor del software simulador

Gracias a la herramienta de simulación Roboguide HandlingPRO, y al interfaz gráfico software, podemos realizar la programación del robot siguiendo los menús de la aplicación con un entorno mucho más cómodo y visual que la programación mediante TPE. En el siguiente apartado se va a comentar todo lo referente a la utilización del interfaz de simulación, funcionalidades y en definitiva a la programación offline de un proceso robotizado utilizando la herramienta de simulación HandlingPRO® del software Roboguide®.

Las configuraciones principales del robot se pueden llevar a cabo desde la aplicación utilizando principalmente la barra de menús principales, las barras de herramientas y el menú "Cell Browser".

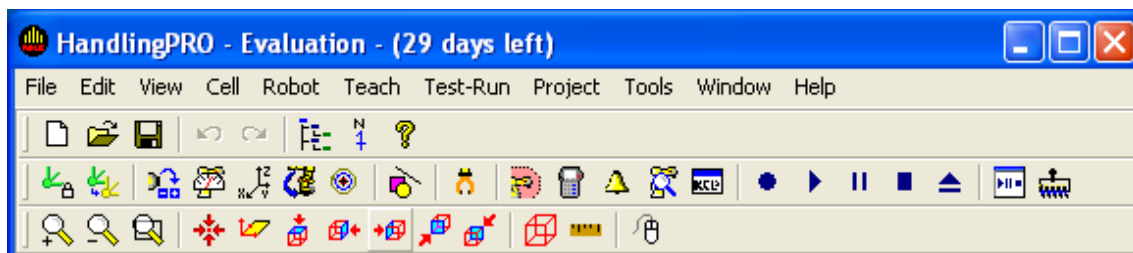


Fig. 38. Barras de acceso rápido del software HandlingPRO Roboguide®

En la primera barra (menús principales) podemos distinguir los menús principales de los que consta la aplicación (*File, Edit, View, Cell, Robot, Teach, Test-Run, Project, Tools, Window y Help*), cuyos nombres indican el menú al que podemos acceder. En cada uno de ellos podremos ver las distintas opciones de las que constan.

La segunda barra nos muestra las opciones de archivo, los botones de "Undo" y "Redo" y por último la visibilidad del "Cell Browser" y "Navigator", además del acceso a la ayuda de la ventana actual.

La tercera barra nos muestra los botones de acceso rápido a las funcionalidades más utilizadas y a las barras rápidas más útiles (*Teach tool selection, Moveto Retry, Jog Coordinates Quick Bar, Gen Override Quick Bar, Teach Quick Bar, Moveto Quick Bar, Target tools*). También disponemos de los botones de interacción con el entorno gráfico (*Draw features on parts, Hand, Work Envelope, Teach Pendant, Robot Alarms, Robot Browser y KCL*). El siguiente conjunto de botones nos permite controlar la ejecución de nuestra simulación y incluso generar un archivo de video AVI (*Record, Play, Pause, Stop, Fault Reset, Fault Reset y Position Edit*).

Utilización de barra de acceso rápido:

A continuación se detalla cada uno de los botones rápidos y menús a los que estos acceden en la aplicación:

File Options



- Permiten crear, abrir o cargar un archivo de simulación.

Undo / Redo



- Permiten deshacer y rehacer las acciones llevadas a cabo en la simulación.

Cell Browser



- Menú principal donde configurar cada uno de los objetos de la simulación.

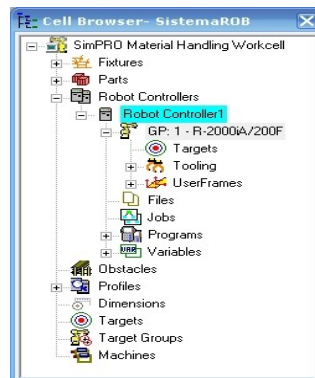


Fig. 39. Imagen del menú "Cell Browser".

Navigator



- Este menú nos indica paso a paso cada una de las acciones para poder programar una simulación.

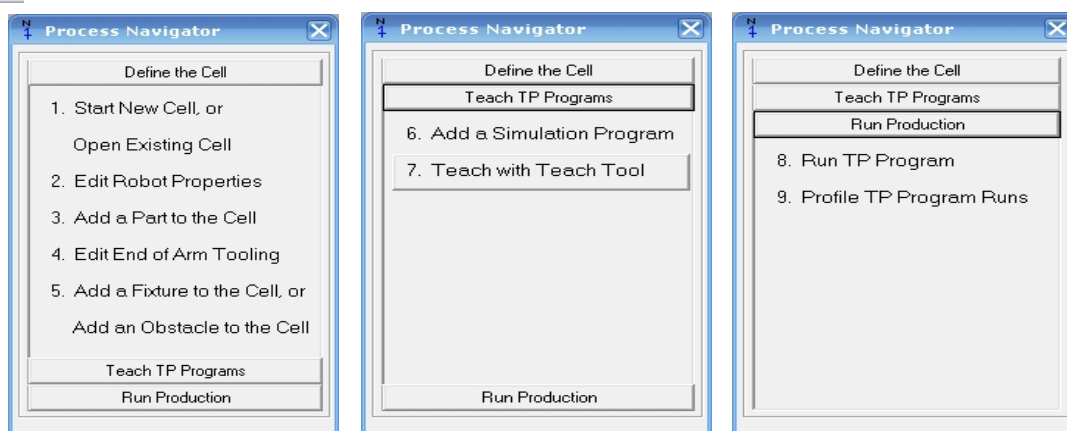


Fig. 40. Imágenes del menú "Navigator".

Help current window



- Muestra el menú de ayuda de la ventana en la que estemos trabajando en un momento determinado.

Teach Tool Selection



- Permite bloquear o desbloquear la utilización de la herramienta de aprendizaje mediante selección del entorno gráfico. Facilita el uso de la herramienta ya que se desactivan el resto de objetos del entorno. Este método es equivalente a la definición de puntos mediante "teach by nose".

Move to Retry



- Activa o desactiva el reintento de la función "MoveTo" si esta falla

Jog Coordinates Quick Bar



- Muestra u oculta la barra rápida del sistema de coordenadas del movimiento del robot.

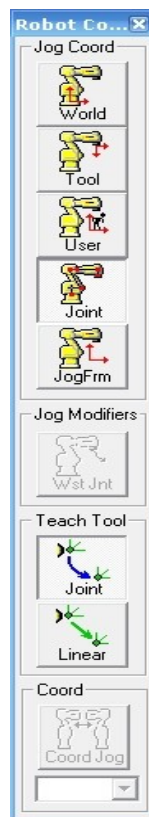


Fig. 41. Imagen del menú "Jog Coodinates Quick Bar".

Override Quick Bar



- Muestra u oculta la barra de configuración de la velocidad override.

Teach Quick Bar



- Muestra u oculta la barra de aprendizaje rápido.

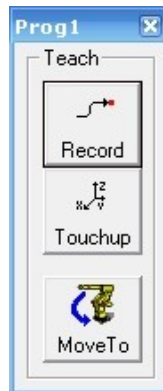


Fig. 42. Imagen del menú "Teach Quick Bar".

MoveTo Quick Bar



- Muestra u oculta la barra rápida de movimiento del robot a determinadas superficies, aristas, vértices...



Fig. 43. Imagen del menú "MoveTo Quick Bar".

Target Tools



- Muestra la barra de herramientas de la configuración de las distintas posiciones (objetivos) a configurar.

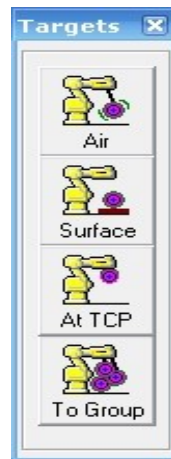


Fig. 44. Imagen del menú "Target Tools".

Draw features on parts



- Permite dibujar distintas líneas, formas y superficies en algunos objetos de la simulación.

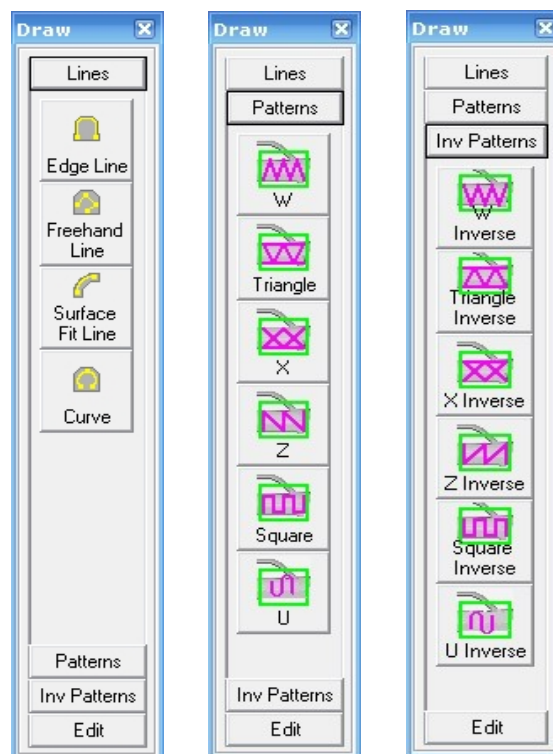


Fig. 45. Imágenes de los menús "Draw features on parts".

Open/Close Hand



- Nos ofrece la posibilidad de abrir y cerrar la herramienta configurada en el TCP del robot.

Work Envelope



- Muestra el alcance o rango de movimiento permitido para el robot con el que estamos realizando la simulación.

Teach Pendant



- Muestra u oculta el Teach Pendant Virtual que proporciona la herramienta de simulación HandlingPRO.

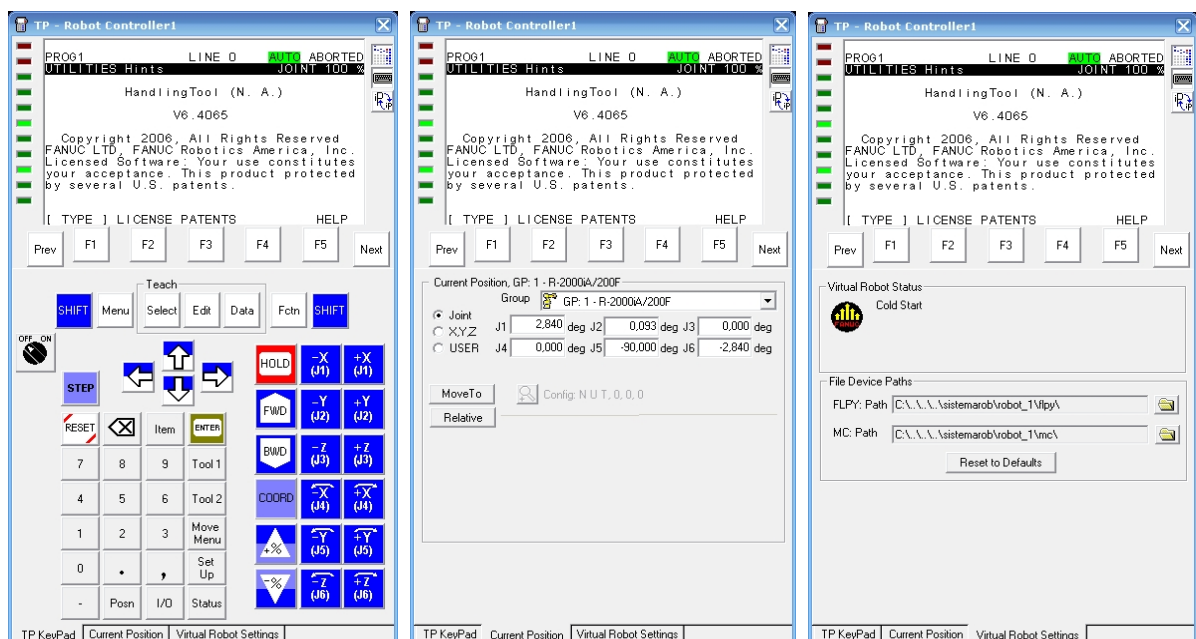


Fig. 46. Imágenes de los menús "Teach Pendant".

Robot Alarms



- Muestra u oculta la consola de alarmas del robot.

#	Alarm Text	Code	Severity	Time
36	MotorSpd lim/DVC[G:1 A:4]	SRVO-171		23:49:48 31/05/2011
37	MotorSpd lim/DVC[G:1 A:3]	SRVO-171		23:49:48 31/05/2011
38	Speed limits used [G:1]	MOTN-056		23:49:48 31/05/2011
39	HOLD signal from SOP/UP is lost	SYST-034		23:47:44 31/05/2011
	Reset	Reset	Reset	Reset
41	Position not reachable	MOTN-018		23:47:26 31/05/2011
	Reset	Reset	Reset	Reset

Fig. 47. Imagen del menú "Robot Alarms".

Robot Browser



- Muestra u oculta el navegador del robot para mostrar información general, de entrada y salida y de monitorización de algunos indicadores del controlador.

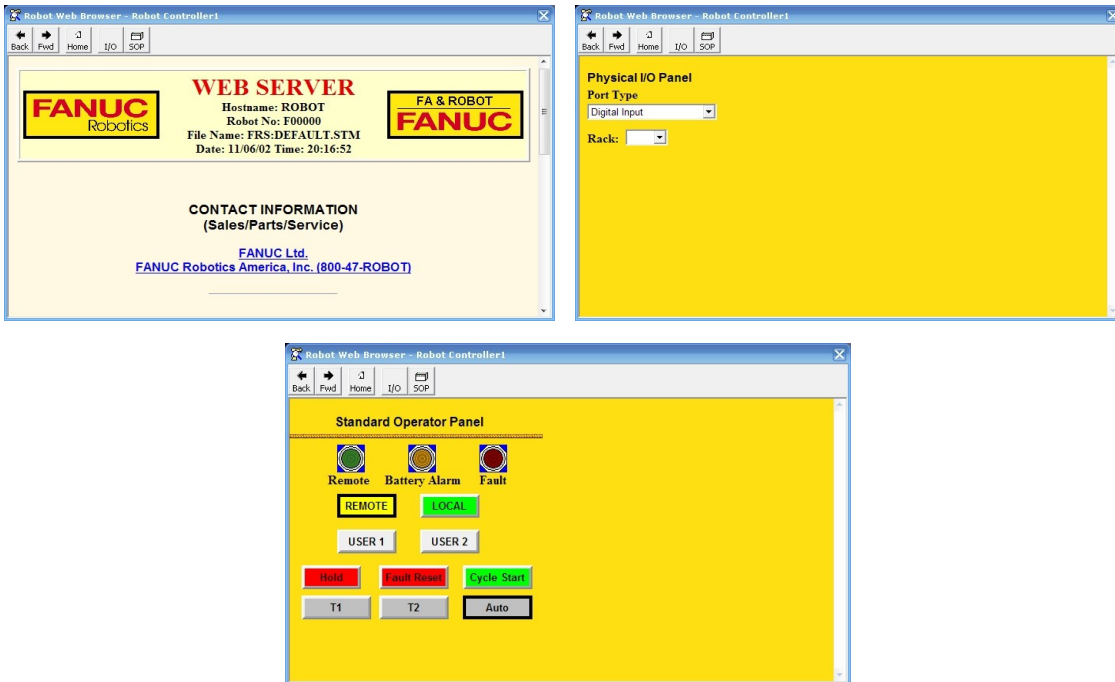


Fig. 48. Imágenes de los menús "Robot Browser".

Record AVI



- Posibilita obtener un video en formato AVI de la simulación.

Cycle Start



- Inicia el ciclo de la simulación con la ejecución secuencial de las instrucciones.

Hold



- Pausa la simulación.

Abort



- Aborta la simulación del proceso.

Fault Reset



- Resetea el conjunto de errores obtenidos en la ejecución de la simulación.

Run Panel



- Muestra u oculta el panel de ejecución de la simulación.

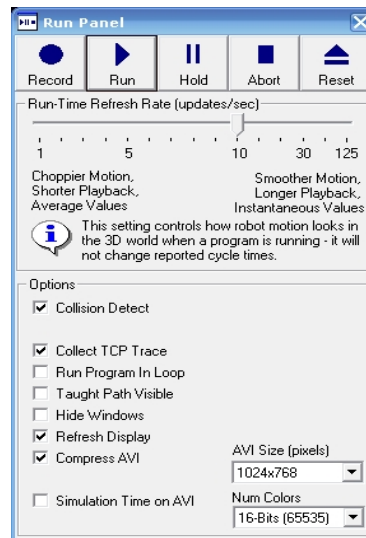


Fig. 49. Imagen del menú "Run Panel".

Position Edit



- Muestra la ventana de edición de posición de los distintos puntos definidos en las instrucciones. Permite también editarlos y redefinirlos de una forma cómoda.

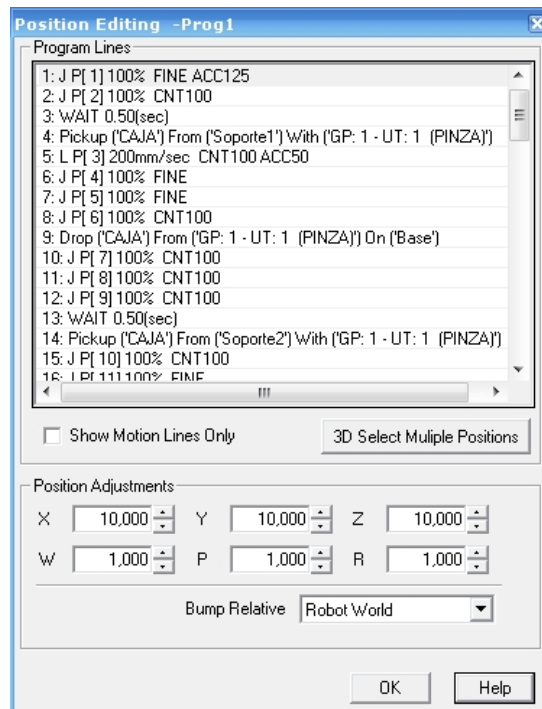


Fig. 50. Imagen del menú "Position Edit".

Zoom Options



- Ofrece las opciones de ampliar vista, alejar vista y *zoom* de ventana.

Center View Selected Object



- Centra en pantalla el objeto seleccionado.

Change Camera Direction



- Posibilita el cambio de la cámara en la dirección de los tres ejes del sistema de coordenadas del entorno.

View Options



- Ofrece la opción de cambiar las vistas frontal, laterales, trasera y cenital del entorno de la simulación.

View WireFrame



- Cambia la presentación de la simulación mostrando las aristas de los objetos.

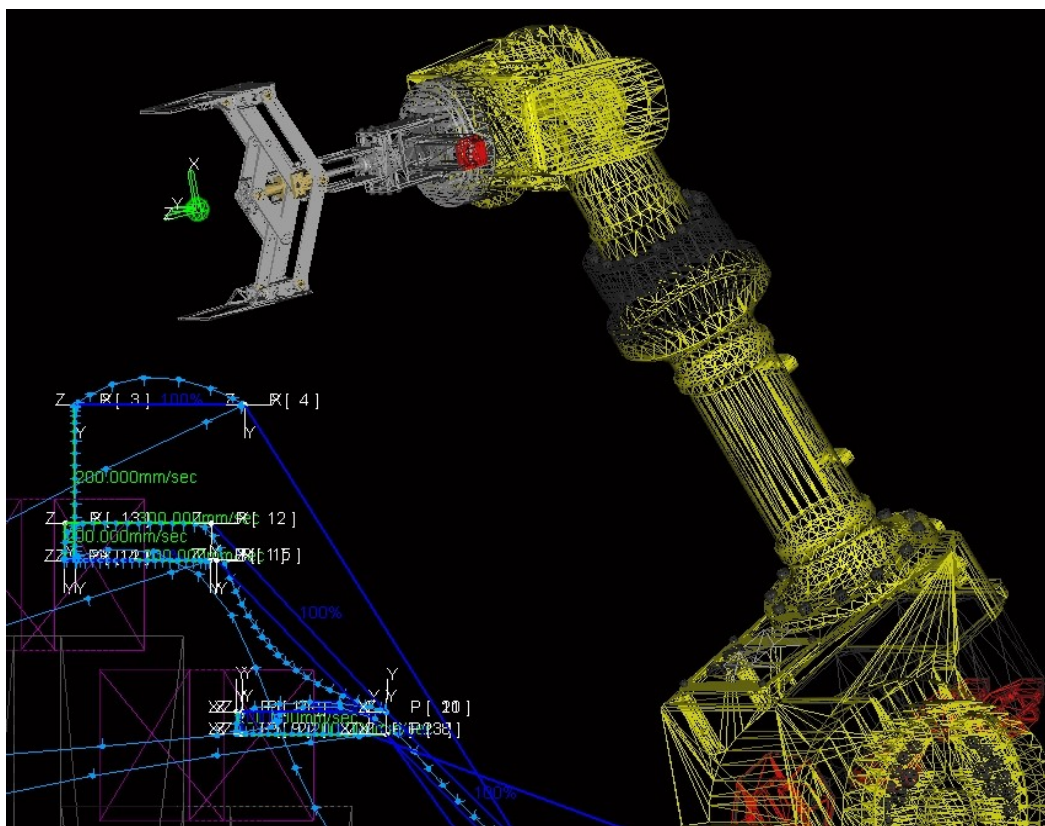


Fig. 51. Imagen de la vista "View WireFrame".

MeasureTool



- Muestra el menú-herramienta de medida.

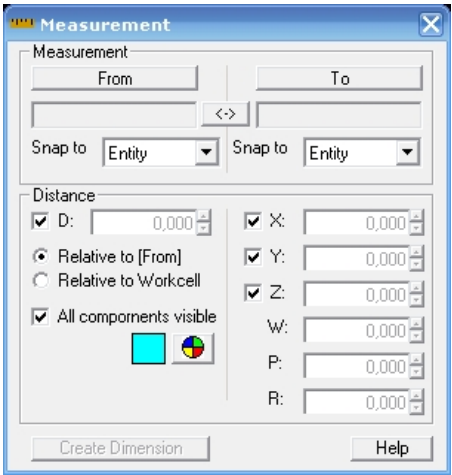


Fig. 52. Imagen del menú "Measure Tool".

Mouse Commands



- Muestra u oculta las combinaciones especiales y los comandos para el ratón.

Mouse Commands					
Function	Action	Function	Action	Function	Action
Rotate view:	RIGHT Drag	Object property page:	DOUBLE-LEFT Click	Move robot to surface:	[CTRL] + [SHIFT] + LEFT-Click
Pan view:	[CTRL] + RIGHT Drag	Move object, one axis:	LEFT Drag triad axis	Move robot to edge:	[CTRL] + [ALT] + LEFT-Click
Zoom in/out:	BOTH Drag (mouse Y axis)	Move object, multiple axes:	[CTRL] + LEFT Drag triad	Move robot to vertex:	[CTRL] + [ALT] + [SHIFT] + LEFT-Click
Select object:	LEFT-Click	Rotate object:	[SHIFT] + LEFT Drag triad axis	Move robot to center:	[SHIFT] + [ALT] + LEFT-Click

Fig. 53. Imagen del menú "Mouse Commands".

Una vez especificados los botones de acceso rápido, describiremos las funcionalidades de la aplicación Roboguide® accesibles mediante los menús principales de la aplicación. Éstos nos permiten realizar todas las funciones del aplicativo de simulación, además de las funciones de los botones de la barra de acceso rápido.

A continuación se describirán los menús principales de la aplicación y se describirán también las acciones que permiten realizar en la simulación:

Descripción de menús principales:

- **File** Es el menú que permite gestionar los archivos de las simulaciones. Desde él podemos manejar las opciones de crear una nueva simulación, abrir un trabajo guardado de simulación, guardar el trabajo realizado de la simulación actual y demás tareas de gestión de los archivos. Destacamos entre ellas la opción de exportar el trabajo al estándar de datos IGES de los programas de diseño CAD mediante la opción "Export", generando así un archivo de tipo de extensión ".igs" para ser importada en una aplicación CAD. Cabe destacar entre las opciones la de "Package cell" que es la utilidad que deberemos usar para poder llevar un trabajo de simulación de un equipo a otro con Roboguide® instalado, ya que las opciones de "restore save point" "save cell" y "open cell" son para los trabajos de simulación del mismo equipo. Para cargar este paquete de la simulación en un nuevo equipo, podremos hacerlo mediante un doble *click* sobre el archivo ".frw" (en nuestro caso SistemaROB.frw).

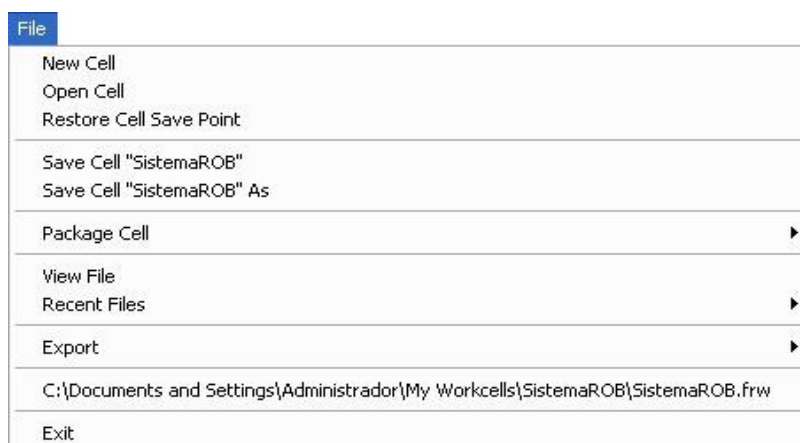


Fig. 54. Imagen del menú principal "File".

- **Edit** Es el desplegable de edición de objetos estándar. Las opciones que ofrece son las de "undo", "redo", "cut", "copy", "paste" y "delete" para aplicarlas a cualquier objeto del entorno de la simulación.



Fig. 55. Imagen del menú principal "Edit".

- **View** En este menú se muestran todas las opciones de visualización del entorno de simulación. Podemos configurar los modos de vista descritos anteriormente en los botones del acceso rápido, además de las barras de acceso de rápido de configuración de movimiento. Destacamos la opción "Program Details" que nos ofrece distintas opciones de visualizado de detalles referentes al robot, EOATs configuradas y opciones de movimiento adicionales.

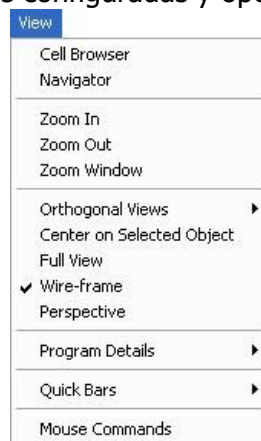


Fig. 56. Imagen del menú principal "View".

- **Cell** El menú contiene todas las opciones que tienen relación con la simulación (célula de trabajo) y los objetos que ésta puede llegar a tener. Desde este menú podemos añadir los mencionados objetos, tales como robots, elementos fijos, partes móviles, obstáculos y objetivos. En este conjunto de opciones, al definir un nuevo objeto podemos escoger cargar un archivo de diseño CAD de la librería predefinida, diseños CAD personalizados y realizados por el usuario y cuerpos geométricos tipo, como cubos, cilindros y esferas.

Destacamos entre las opciones las "I/O interconnectons" que permiten interconectar las interfaces de entrada y salida entre robots, y las "Workcell Properties" que muestran propiedades generales de configuración de la célula.

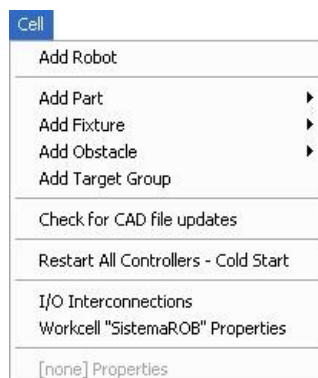


Fig. 57. Imagen del menú principal "Cell".

- **Robot** El menú nos indica las opciones que podemos elegir para manejar el robot en cuestión. Tenemos acceso a la consola "*Teach pendant*" para realizar esta configuración, además de la utilización del resto de menús y de la "*teach tool*".

La "*teach tool*" es la principal herramienta para el movimiento del TCP, que nos permitirá la definición de puntos en el espacio y alcance de objetivos de una forma sencilla para luego poder obtener las coordenadas y ajustar el movimiento con mayor precisión. Para ello podremos usar el interfaz TP o el menú de definición de programa "*Teach prog1 program*", el cual explicaremos en el siguiente punto.

Podremos encontrar varios de los botones de acceso rápido, y como opción destacable la de "Robot GP1:modelo Properties", que nos permitirá acceder a opciones de configuración del robot seleccionado.



Fig. 58. Imagen del menú principal "Robot".

- **Teach** El menú "teach" nos muestra las opciones de las que disponemos para realizar la programación de los movimientos y acciones que el robot puede llevar a cabo en los distintos programas de la simulación. Podemos añadir un programa de simulación, un programa mediante la consola "teach pendant", o cargar un programa ya definido.



Fig. 59. Imagen del menú principal "View".

También tenemos acceso al menú "draw part features" descrito anteriormente en las barras de acceso rápido, además de dos nuevos menús, "Teach Program ProgX" que nos permitirá acceder al menú de programación principal, y "Program ProgX Properties" que nos muestra las opciones de configuración del Programa seleccionado.

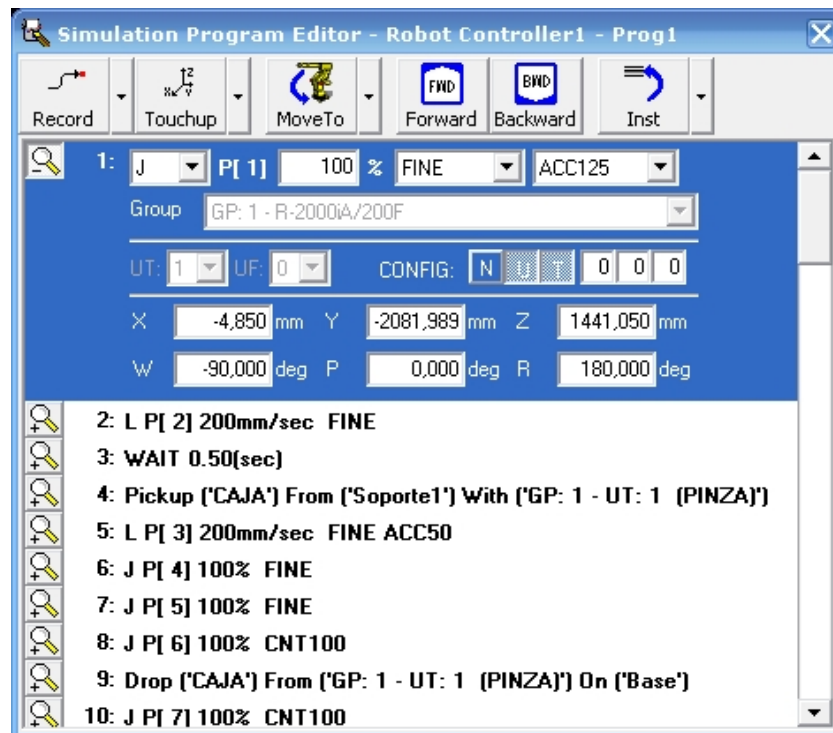


Fig. 60. Imagen del menú "Teach Program".

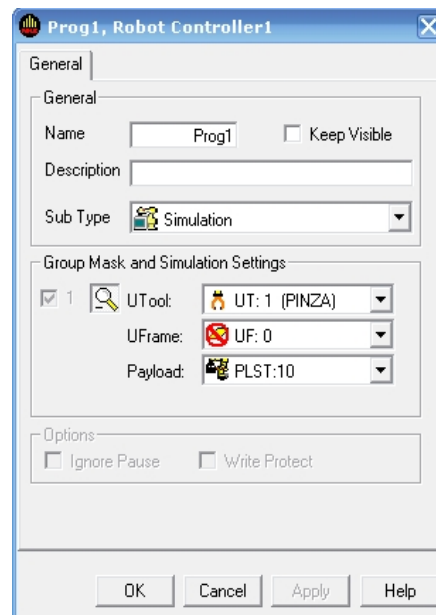


Fig. 61. Imagen del menú "Program Properties".

- **Test-run** El desplegable nos ofrece las distintas opciones de ejecución de simulación. Podemos acceder al panel de ejecución, configuración de ejecución y a las opciones de ejecución.



Fig. 62. Imagen del menú principal "Test-run".

Entre las opciones podemos encontrar algunas funcionalidades, de entre las cuales podemos destacar la de "Colision Detect", que analiza las posibles colisiones que se puedan producción durante la ejecución de la simulación.

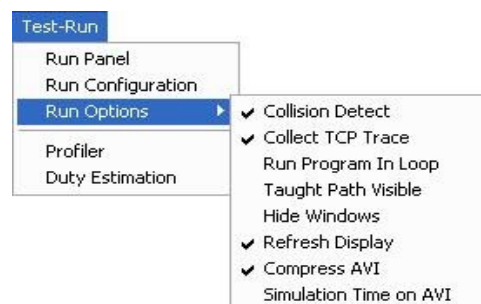


Fig. 63. Imagen del menú principal "Test-run" > "Run Options".

- **Project** El menú nos ofrece una forma alternativa para acceder a las opciones de desarrollo para la controladora utilizada. Las opciones que aparecen son equivalentes a los menús de botón derecho de los botones de desarrollo del "Cell Browser". Destacamos las opciones "Export" e "Import", que nos permiten exportar e importar masivamente archivos de simulación, programas y configuraciones del sistema a la aplicación de simulación o al robot de producción.

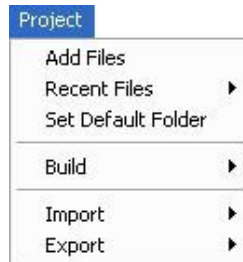


Fig. 64. Imagen del menú principal "Project".

- **Tools** Nos permite acceder al directorio predeterminado del robot seleccionado para tratar los archivos asociados, y nos da acceso también a un menú de configuración de opciones generales, tales como la creación de plantillas en sentencias de definición de puntos, tiempo de refresco de la imagen en "teach time", configuración de colores de objetos, definición de rutas de trabajo de la aplicación,...



Fig. 65. Imagen del menú principal "Tools".

- **Window** Nos ofrece la posibilidad de cambiar los tipos de interfaz de la aplicación, pudiendo definir u interfaz gráfico más sencillo o más complejo, a través de la opción "3D Panes".

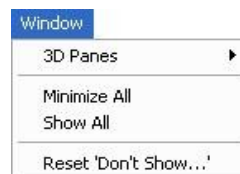


Fig. 66. Imagen del menú principal "Window".

- **Help** Muestra las distintas opciones de ayuda que ofrece el programa.



Fig. 67. Imagen del menú principal "Help".

5. Simulación del sistema robótico.

5.1 Diseño de la célula de trabajo.

Para la realización del diseño del proceso tendremos en cuenta las especificaciones definidas en el apartado 3.2, componiendo una célula de trabajo tipo, característica de un entorno industrial productivo.

Se definen varias zonas dentro del sistema, por las cuales los objetos manipulados van a ser trasladados según el estado del proceso robótico. Denotamos pues, cuatro zonas diferenciadas que son:

- *Posición del robot manipulador.* Se definirá en esta posición el origen de coordenadas para el sistema robotizado.
- *La zona de soporte de cajas.* Se proveerán cajas a través de un sistema automático en el soporte 1, y mediante la acción del manipulador se irán intercalando las posiciones de las mismas en los tres niveles a modo de cola FIFO.
- *La zona de soportes de la pieza 1.* Se proveerán piezas mediante un sistema automático en el soporte 1 y se retirarán una vez utilizadas, mediante el mismo sistema automático.
- *La zona de trabajo.* Se realizará un trabajo determinado con los objetos trasladados hasta la zona por un robot manipulador.

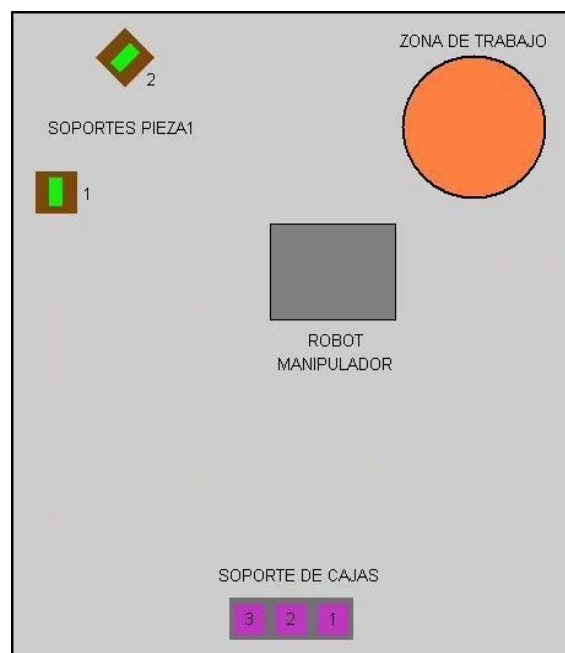


Fig. 68. Diseño de la célula de trabajo del sistema robótico.

5.2 Estados del proceso automático. Red de Petri.

El diagrama de estados del proceso robótico se compone de cuatro acciones disponibles y de siete posiciones admitidas en el "mundo" del manipulador robótico.

Las acciones del proceso son:

- Captura de "Caja"
- Captura de "Pieza1"
- Depósito de "Caja"
- Depósito de "Pieza1"

Las posiciones permitidas en el entorno del manipulador son:

- Soporte 1
- Soporte 2
- Soporte 3
- Base (Zona de trabajo)
- Fixture 1
- Fixture 2

El proceso en sí constará de los siguientes pasos por orden de ejecución:

- Posición inicial.*
- Posicionamiento en "Soporte1".
- Captura de "Caja".
- Posicionamiento en "Base".
- Depósito de "Caja"
- Posicionamiento en "Soporte2".
- Captura de "Caja".
- Posicionamiento en "Soporte1".
- Depósito de "Caja"
- Posicionamiento en "Soporte3".
- Captura de "Caja".
- Posicionamiento en "Soporte2".
- Depósito de "Caja".
- Posicionamiento en "Fixture2"
- Captura de "Pieza1"
- Posicionamiento en "Base".
- Depósito de "Pieza1"
- Posicionamiento en "Fixture1".
- Captura de "Pieza1"
- Posicionamiento en "Fixture2"

- Depósito de "Pieza1".
- Posición de espera.*
- Posicionamiento en "Base".
- Captura de "Caja".
- Posicionamiento en "Soporte3".
- Depósito de "Caja".
- Posicionamiento en "Base".
- Captura de "Pieza1".
- Posicionamiento en "Fixture1".
- Depósito de "Pieza1".
- Posición final.*

Las posiciones marcadas con un asterisco indican que el punto de esa posición coincide exactamente en el espacio definido (inicio, espera y fin).

Siguiendo el flujo de ejecución podemos componer el gráfico de la red de Petri, que se divide en los cuatro estados posibles (acciones de captura y depósito de los objetos) y las 7 transacciones (posicionamiento en los diferentes espacios de manipulación).

A continuación se muestra el mencionado proceso industrial mediante el diseño de la red de Petri:

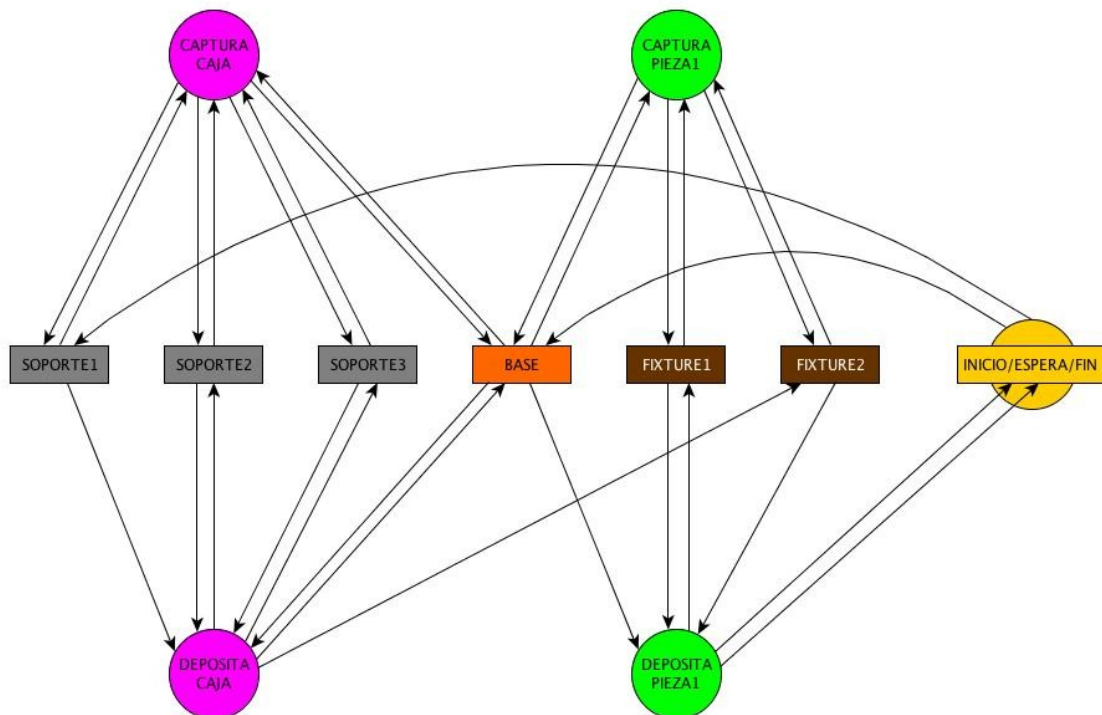


Fig. 69. Diagrama de red de Petri. del sistema robótico.

5.3 *Presentación del manipulador*

Para la ejecución del proceso se ha escogido el manipulador Fanuc R-2000iA/200F.

La serie de robots R-2000iA de Fanuc es la última generación de robots de gran carga y con un alto rendimiento industrial. Podemos destacar de este modelo el rendimiento, la seguridad y la manipulación, asegurando la más alta fiabilidad. El modelo 200F es el modelo estándar, de una amplia gama con pequeñas modificaciones según las necesidades de los procesos a realizar.

Equipado con 6 ejes, siendo la carga máxima es de 200 kg, ofrece un alcance máximo 2650 mm, con una repetibilidad de $\pm 0,2$. ((explicar repetibilidad))

El modelo 200F es el modelo estándar de una amplia gama con pequeñas modificaciones según las necesidades de los procesos a realizar.

Gracias a la capacidad de carga útil de este modelo de robot y a los grados de libertad que proporcionan los 6 ejes de los que dispone tenemos un manipulador que se adapta perfectamente al proceso que deseamos implementar. Estas características hacen de este modelo un robot muy versátil para alcanzar un amplio número de trayectorias y dota al sistema de una gran capacidad para poder ser ampliado sin repercusiones de importancia.



Fig. 70. Robot FANUC R2000iA Series.

En las siguientes imágenes se muestran los planos del robot con las medidas generales especificadas por distintas vistas y detalles del robot R2000iA/200F.

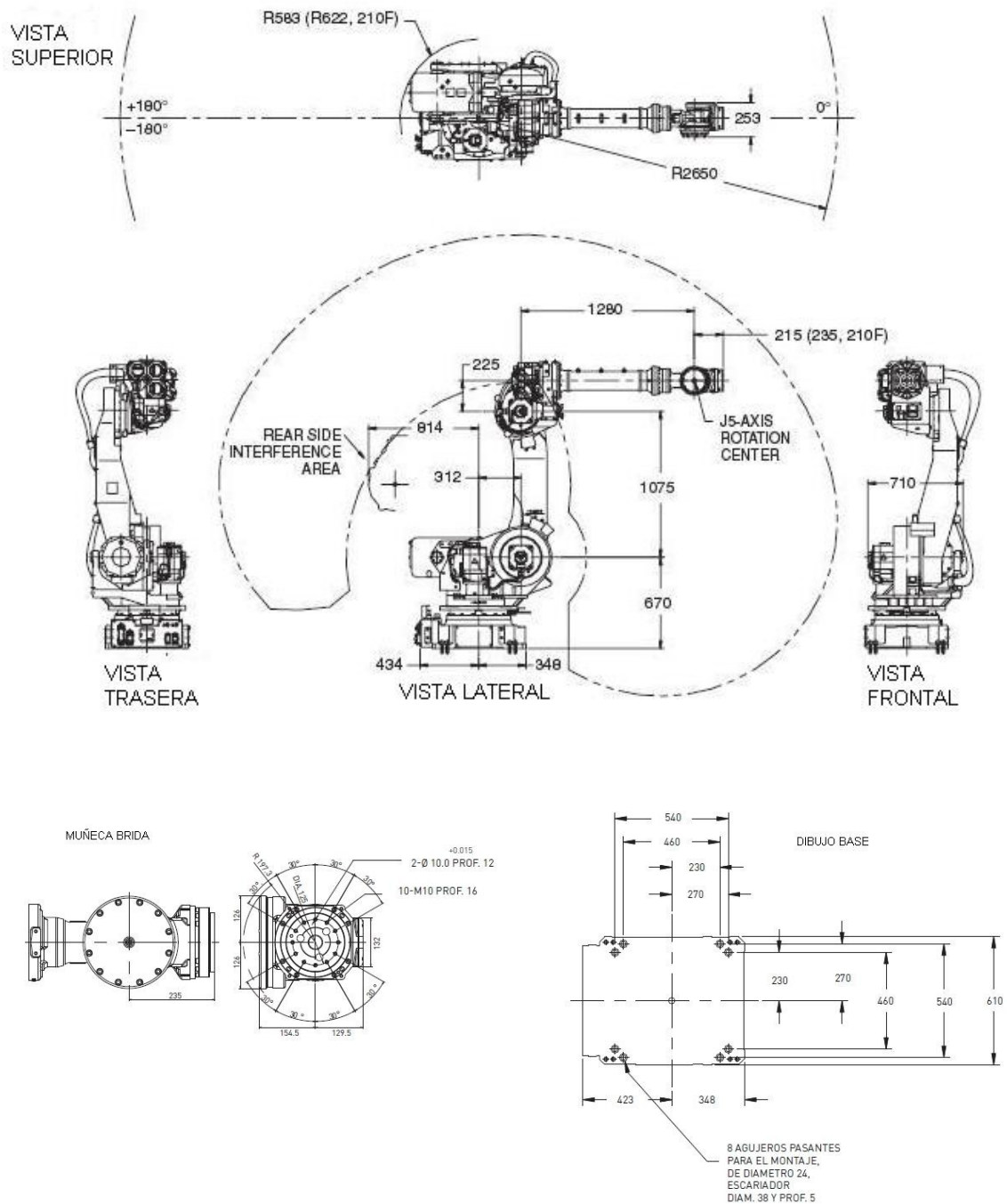


Fig. 71. Planos del robot FANUC R-2000iA/200F.

5.4 Implementación del sistema automático.

Una vez especificado el proceso industrial a programar, desarrollaremos el entorno virtual de simulación por software Roboguide® HandlingPRO® de FANUC Robotics.

Para el desarrollo del proyecto de simulación utilizaremos la versión de evaluación 6.40 (Rev. E) para Windows XP.

5.4.1 Definición y configuración del robot y la controladora.

Para comenzar a utilizar la aplicación, una vez instalada en el sistema operativo deberemos de crear una nueva célula de trabajo. Podemos utilizar el menú *File > New Cell*, o bien utilizar el menú "navigator", que nos guiará paso a paso para comenzar con nuestra simulación 3D.

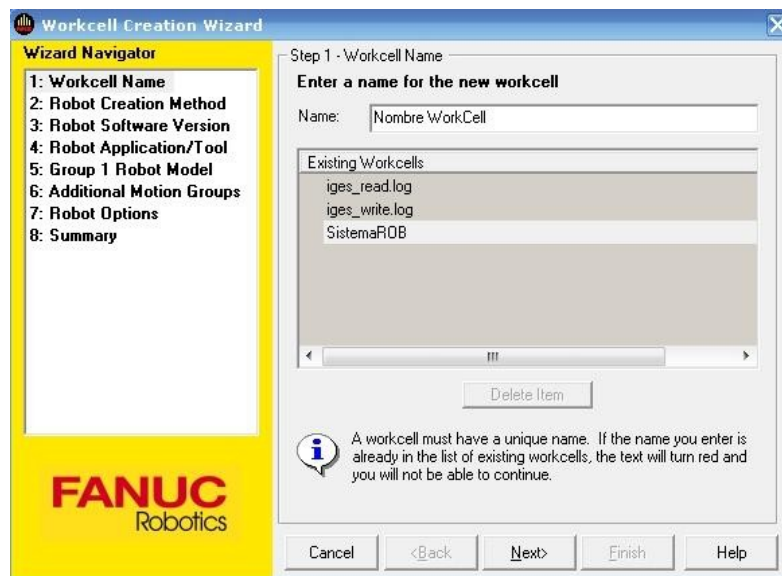


Fig. 72. Asistente de creación de célula de trabajo "Step 1".

Desde esta pantalla tenemos la opción de poder eliminar las células de trabajo que no sean necesarias, una vez realizada su programación y simulación.

Daremos un nombre a la "WorkCell" y pasaremos a la siguiente ventana del menú. En esta pantalla podremos elegir entre: crear un nuevo robot utilizando la configuración por defecto de HandlingPRO; crear un robot con la última configuración utilizada; crear un robot a partir de un archivo de *backup* o hacer una copia de un trabajo ya existente.

Utilizaremos la primera opción, ya que nos permite modificar las opciones de configuración *a posteriori* si fuese necesario para la simulación.

En la siguiente figura se muestra la pantalla del asistente referente a este punto:

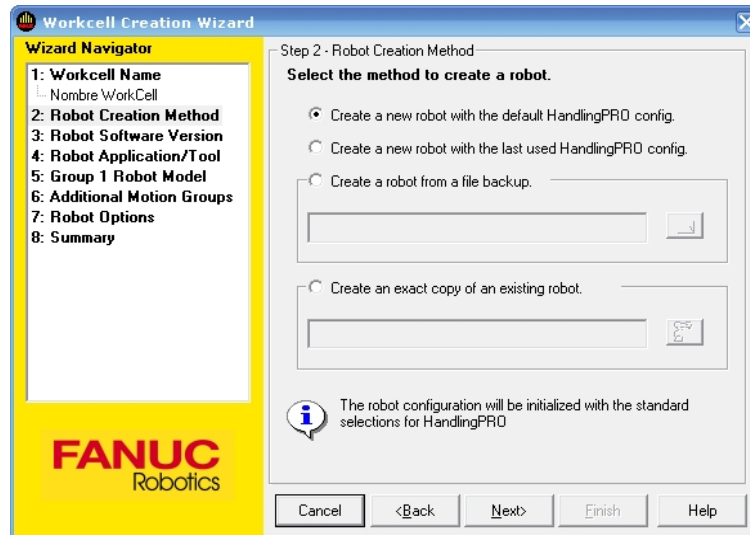


Fig. 73. Asistente de creación de célula de trabajo "Step 2".

En la tercera pantalla del asistente es necesario escoger la versión del software de la controladora del robot ofrecido por la aplicación. Disponemos de distintas versiones desarrolladas por el fabricante FANUC. Cuando se realiza la instalación de la aplicación se solicitan las versiones de software de la controladora que se desean instalar, de las cuales hemos configurado las versiones 6.40 y 7.20. Para realizar nuestra simulación hemos elegido la versión 6.40.

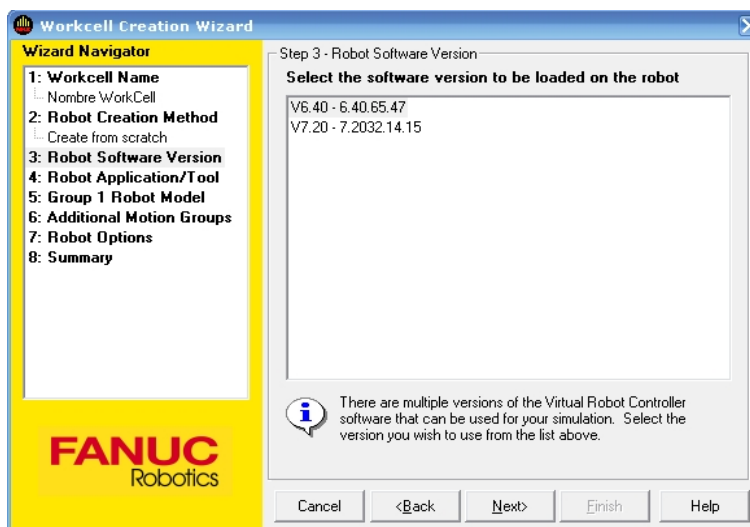


Fig. 74. Asistente de creación de célula de trabajo "Step 3".

En la pantalla que se muestra a continuación podemos escoger el paquete de la aplicación que vamos a utilizar. En este caso se trata del paquete "HandlingTool" (H552), que nos permitirá realizar la simulación de un proceso de "pick and place"

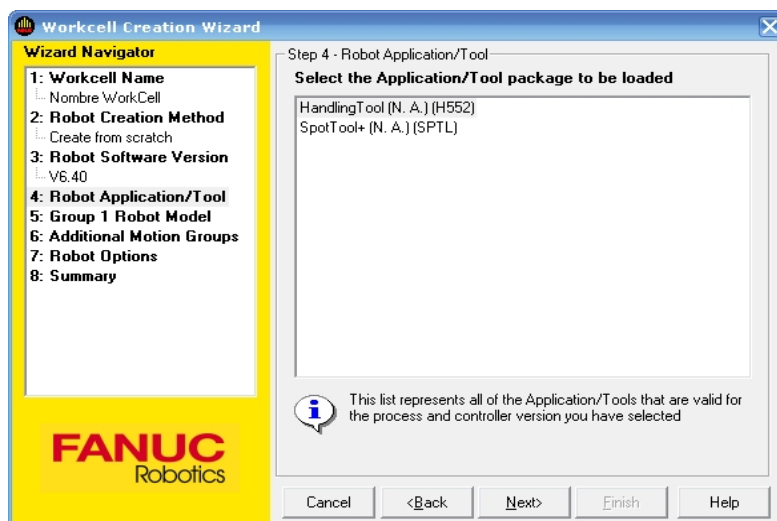


Fig. 75. Asistente de creación de célula de trabajo "Step 4".

En el paso número cinco del asistente podemos elegir el robot que vamos a utilizar para realizar el proceso aplicado a la planta industrial. Tras el estudio previo de las necesidades físicas escogeremos el modelo que mejor se adapta a nuestros requisitos.

Hemos elegido el modelo de robot R2000iA/200F por tener una buena capacidad de carga y ser uno de los más versátiles en cuanto a movilidad.

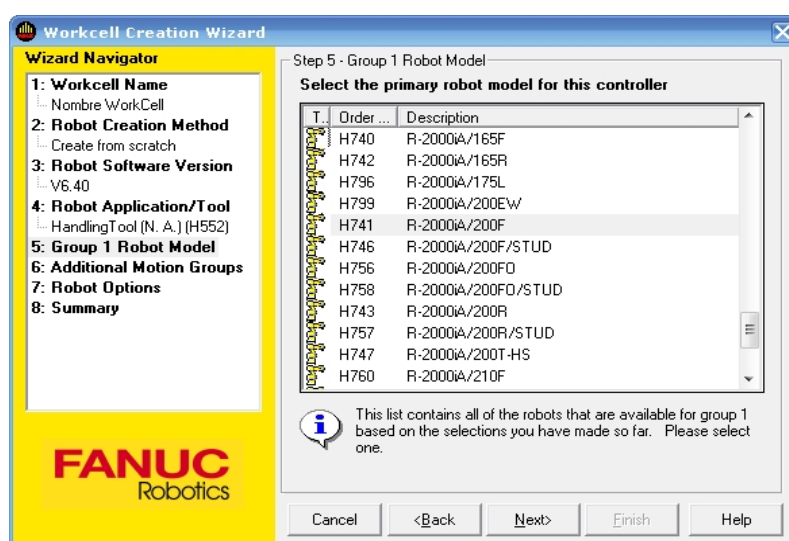


Fig. 76. Asistente de creación de célula de trabajo "Step 5".

En el siguiente paso, la aplicación nos permite definir “grupos de movimiento”, añadiendo más robots a la célula además del ya seleccionado. Incluso nos permite crear varios grupos de movimiento con distintos robots en cada uno de ellos.

Esta opción puede reconfigurarse una vez en la aplicación y según las necesidades, por lo que hemos seleccionado un único robot para poder añadirlo posteriormente desde la aplicación.

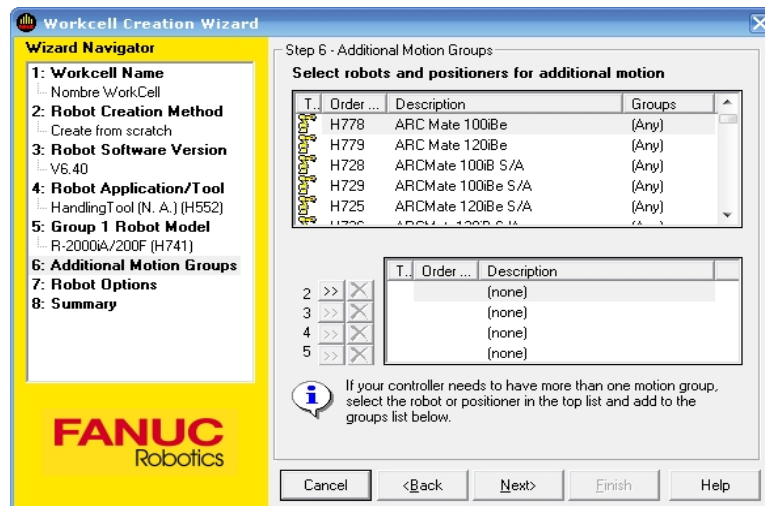


Fig. 77. Asistente de creación de célula de trabajo "Step 6".

A continuación definiremos los módulos de software específicos de FANUC que cargaremos directamente en la controladora del robot. Son funcionalidades que ofrece este módulo, como por ejemplo: sistemas de visión integrados en el manipulador 2D o 3D, diversos sensores, herramientas para el tratamiento de colisiones, sistemas de medición, máquinas especiales...

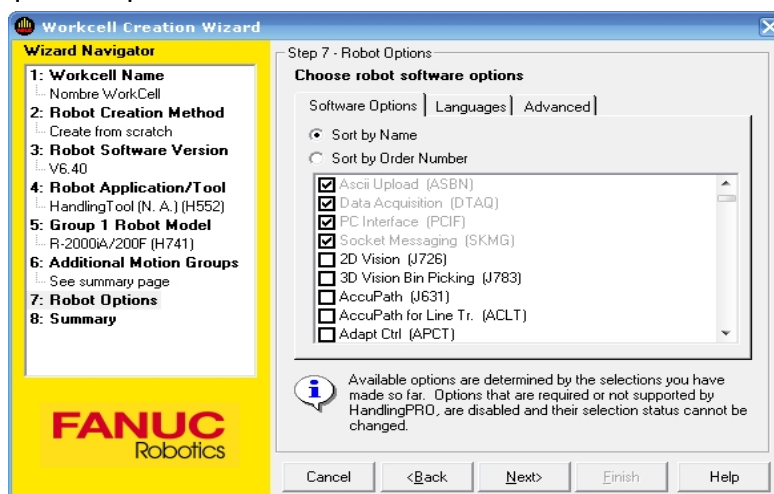


Fig. 78. Asistente de creación de célula de trabajo "Step 7".

En la última pantalla del asistente obtendremos un resumen de las opciones determinadas en las etapas anteriores. Una vez revisados los distintos parámetros, finalizamos la creación del robot en el software de simulación.

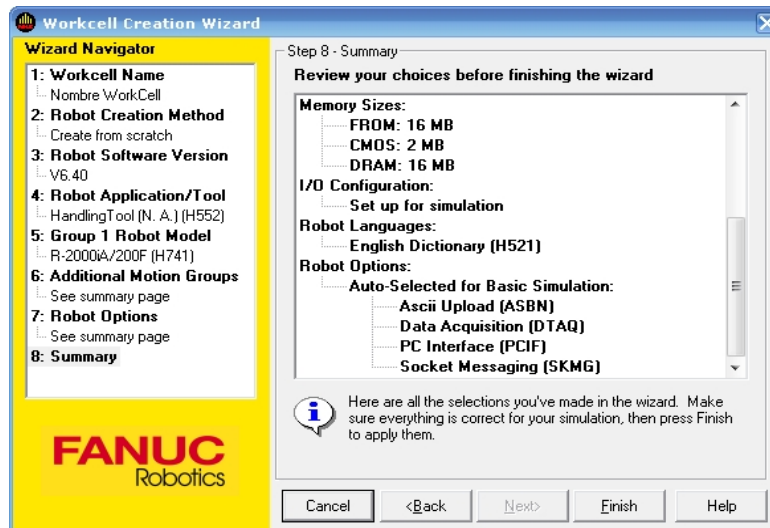


Fig. 79. Asistente de creación de célula de trabajo "Step 8".

Finalmente observamos que el software de simulación inicializa la controladora virtual del robot tal y como se ha configurado. Definirá los valores iniciales de las variables internas del sistema, las funcionalidades cargadas previamente, el "teach pendant" virtual... Una vez terminado el proceso de inicialización las opciones configuradas se encuentran disponibles en el entorno virtual de trabajo 3D.

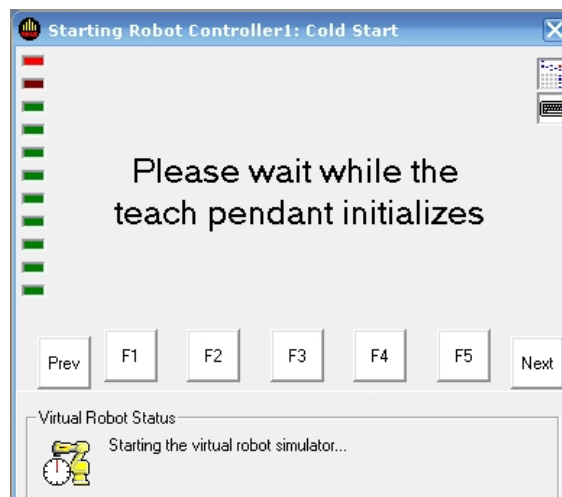


Fig. 80. Inicialización de la controladora del robot.

Después de haber creado el objeto del robot, empezaremos a trabajar con él revisando la configuración de sus propiedades, utilizando el menú accesible con el botón derecho del ratón en la ventana "Cell Browser".

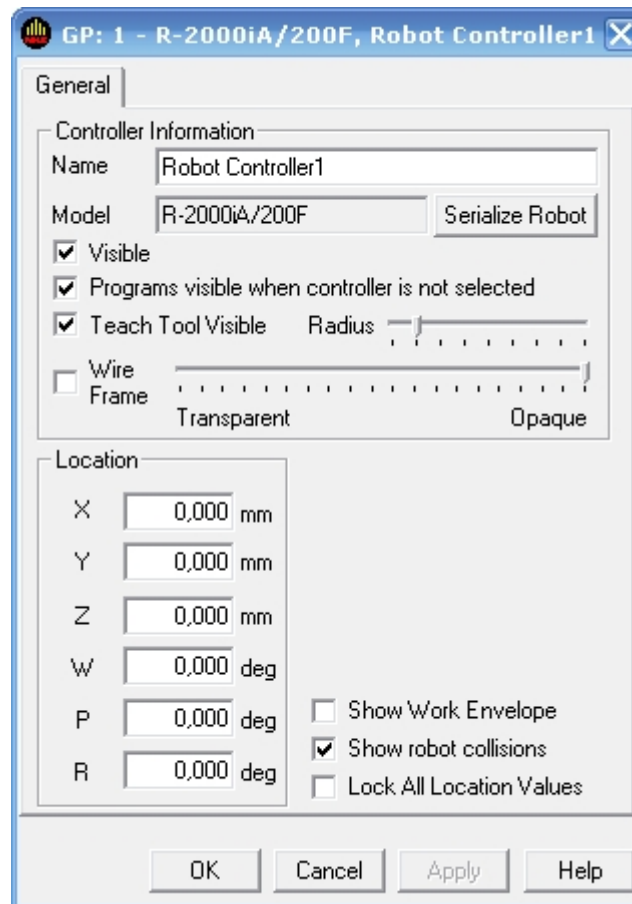


Fig. 81. Menú de las propiedades de configuración del robot R-2000iA/200F.

Como comprobamos en la figura anterior, las propiedades más destacadas del robot que podemos editar son: El nombre del robot, las opciones de visibilidad durante el "teach run", la transparencia u opacidad, la visualización de las colisiones y la posición en el entorno de la célula.

Los pasos que se definen a continuación permitirán configurar los distintos objetos que compondrán el entorno de trabajo para la simulación siguiendo el esquema del diseño de la célula robótica.

5.4.2 Definición de partes y objetos manipulados (PARTS)

Una vez configurado el robot definiremos las propiedades de las piezas que van a ser manipuladas durante la simulación. Para realizar esta acción utilizaremos el menú principal *"Cell Browser"* y sobre el desplegable *"parts"* seleccionaremos, con el botón derecho del ratón, la opción *"add a part"*. El menú nos muestra distintos objetos para cargar como manipulados. También nos permite cargar una parte predefinida de la aplicación, una parte personalizada y distintos cuerpos geométricos (box, cylinder y sphere).

En este caso hemos definido dos partes escogiendo la forma *"box"*, las cuales servirán de ejemplo de objeto manipulado.

Una vez creadas las citadas partes debemos configurarlas con la opción *"Properties"* accediendo de nuevo con el botón derecho desde el menú *"Cell Browser"*.

En la siguiente figura visualizamos la pantalla de configuración de la parte *"CAJA"*.

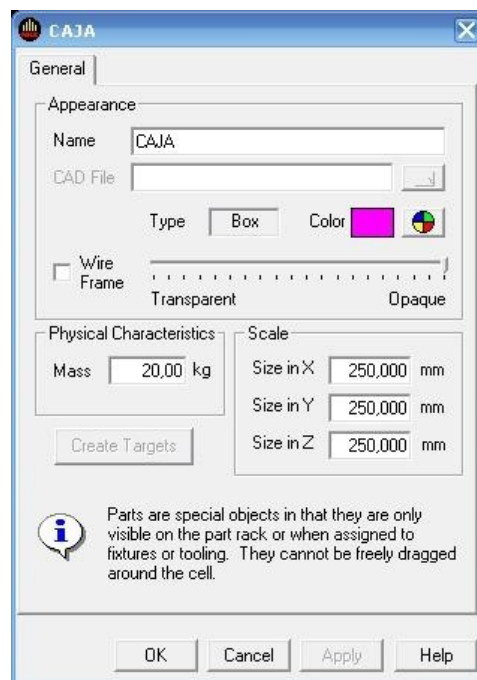


Fig. 82. Menú de propiedades del objeto *"CAJA"* (parts).

Comprobamos que este menú nos permite configurar el nombre del objeto, el archivo CAD en el caso de que carguemos un archivo de imagen CAD, el tipo de objeto (box en este caso), color, opacidad, la masa y el tamaño definido en los ejes x, y, z.

Análogamente definimos el segundo objeto manipulado llamado "pieza1", y accedemos a su ventana de propiedades:

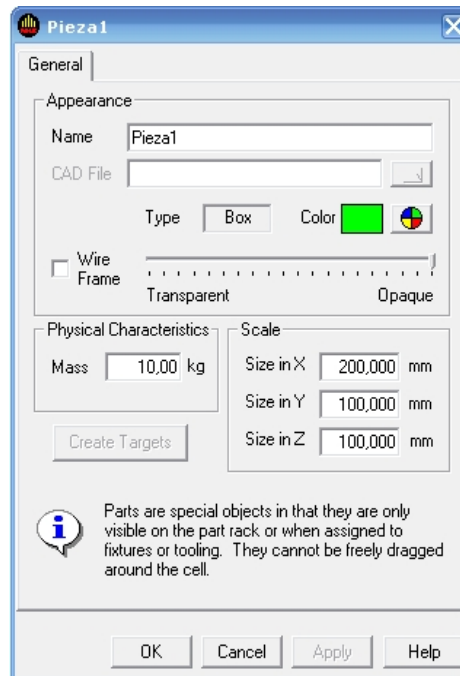


Fig. 83. Menú de propiedades del objeto "CAJA" (parts).

Definidas ya las partes que serán manipuladas en la simulación, la configuración de estos objetos en el "Cell Browser" quedará de la siguiente forma:

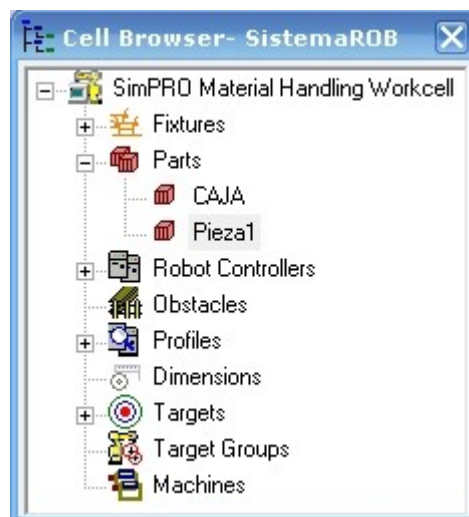


Fig. 84. Definición de los objetos manipulados en la simulación (parts).

5.4.3 Definición de herramienta en TCP (UTOOL)

En este apartado se añadirá una herramienta al TCP del robot para poder manipular los objetos definidos en el menú "parts". Para definir una herramienta EOAT (*End of Arm Tool*) accedemos al menú del "Cell Browser", en la opción "tooling", definiremos para ello una nueva "UTOOL".

Cuando abrimos el desplegable "tooling" comprobamos que el robot permite la definición de distintas "UTOOL" (En el software de simulación hemos encontrado once herramientas posibles).

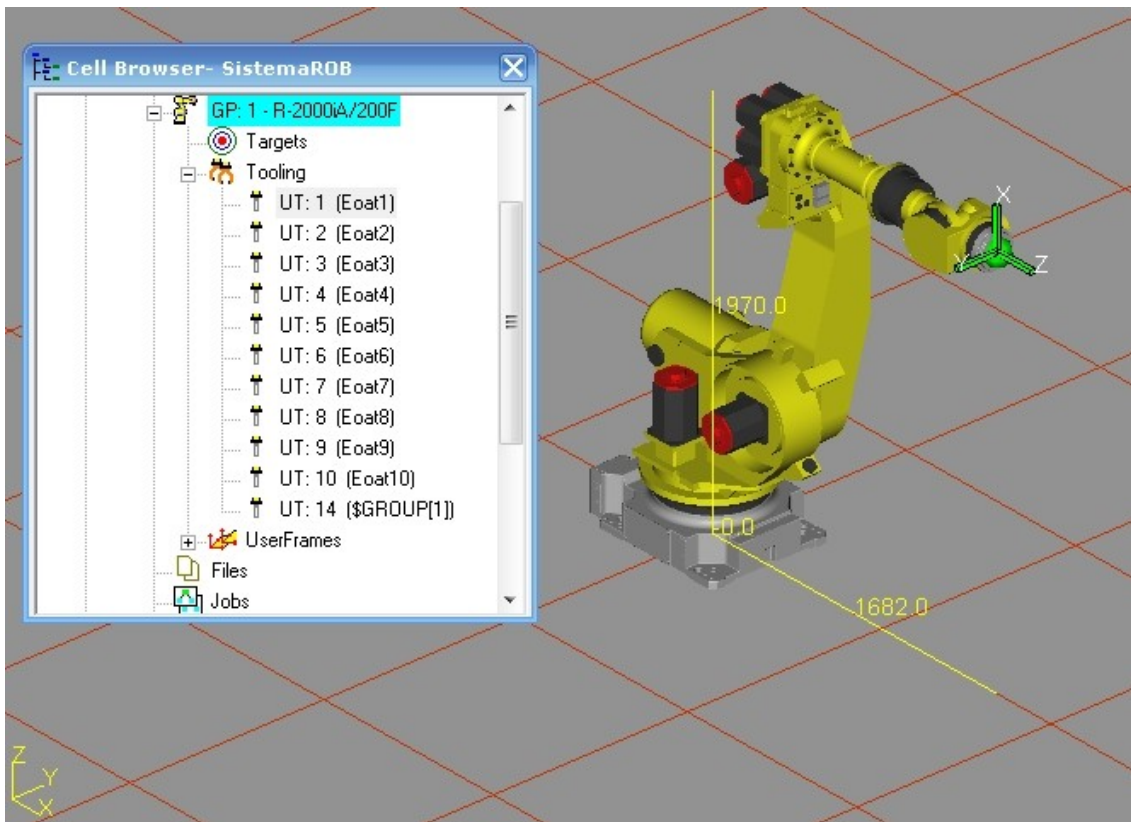


Fig. 85. Menú "tooling" del software de simulación HandlingPRO®.

El procedimiento para definir una herramienta se realiza eligiendo una de las "UTOOL" definidas en el "Cell Browser". El botón derecho del ratón nos permite acceder a la opción "Eoat1 Properties".

Dentro del menú, en la pestaña "General", nombraremos la herramienta a definir, cargaremos el archivo primario CAD de la herramienta (ya que si se trata de una herramienta prensil, tendremos un archivo CAD secundario en estado "cerrado"), posicionaremos la herramienta en el TCP con el desplazamiento deseado ("*offset*"), definiremos la masa de la herramienta y la escala empleada en el entorno gráfico.

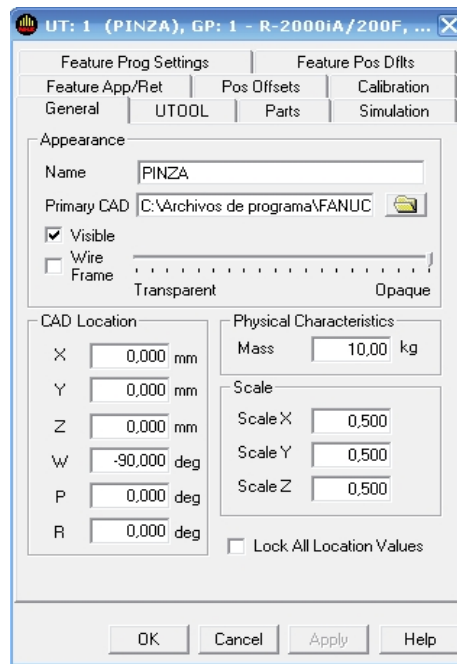


Fig. 86. Definición de características en la pestaña "General".

En la pestaña *UTOOL* definiremos el punto de actuación de la pinza. Debido al desplazamiento en la configuración de la herramienta, definiremos el sistema de referencia del punto de actuación para especificar los movimientos del robot de forma que la pinza pueda posicionarse adecuadamente sobre los objetos a manipular.

Para realizar esta operación insertaremos las coordenadas de forma directa en el sistema de referencia relativo a la pinza o utilizar la funcionalidad "*Use Current Triad Location*" para establecer como sistema de referencia el punto actual definido con la herramienta de desplazamiento del robot.

En esta pestaña de configuración, opcionalmente, también podemos definir la normal que el sistema nos proporcionará por defecto para utilizar la opción de movimiento "*Move Robot Normal-to-surface*" empleando la combinación de teclas Ctrl-Shift-Click. En este caso dejaremos el valor definido para la normal predeterminada en el eje de coordenadas -Z. El método "*current triad location*" se basa en la toma de coordenadas del TCP después de situar el robot con la "*teach tool selection*", equivalente al sistema "*teach by nose*" de un sistema robótico real.

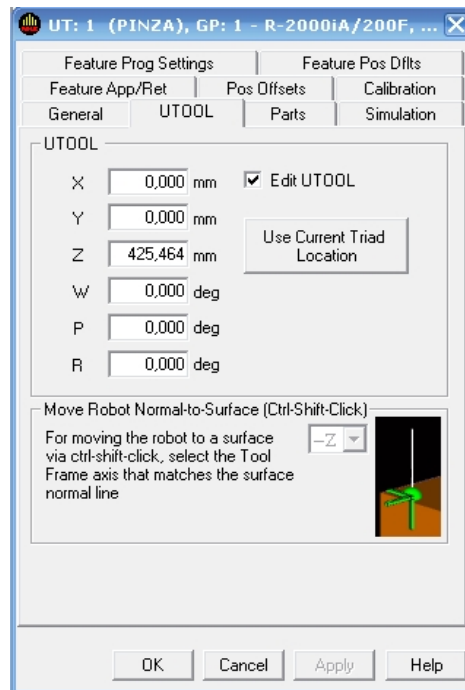


Fig. 87. Definición de características en la pestaña "UTOOL".

En la siguiente pestaña definimos las partes que esta herramienta podrá manipular. Previamente hemos definido las partes en el menú "*parts*" para poder asignarlas a la herramienta que estamos definiendo y que puedan ser manipuladas en la simulación.

Para cada una de las partes definidas debemos definir el "*offset*" cuando la herramienta está actuando sobre ella. En nuestro caso, al tratarse de un proceso de manipulación de piezas hemos de definir el "*offset*" de la pieza cuando ésta esta siendo "agarrada" por la pinza que la va a rotar o trasladar, según el movimiento que se defina en el programa.

El "*offset*" puede definirse en el sistema relativo x, y, z, además de la orientación con los ángulos w, p, r.

Para realizar la edición del "*offset*", debemos marcar la casilla "*Edit part offset*", elegir una de las piezas definidas en el sistema mediante el desplegable del menú, y a continuación podremos introducir los valores de las variables x, y, z, w, p, r.

Podremos definir también si queremos que esta pieza sea visible en el tiempo de aprendizaje "*teach time*" y/o en el tiempo de simulación "*run time*".

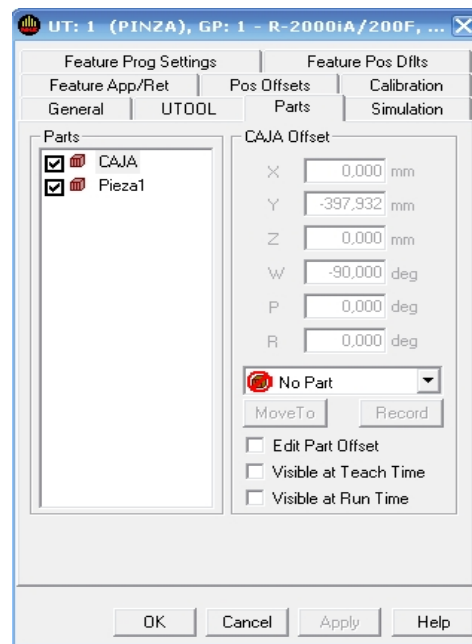


Fig. 88. Definición de características en la pestaña "parts".

Por último definiremos las características de la pestaña "Simulation", lo cual nos permitirá controlar algunos aspectos de la herramienta en el tiempo de simulación.

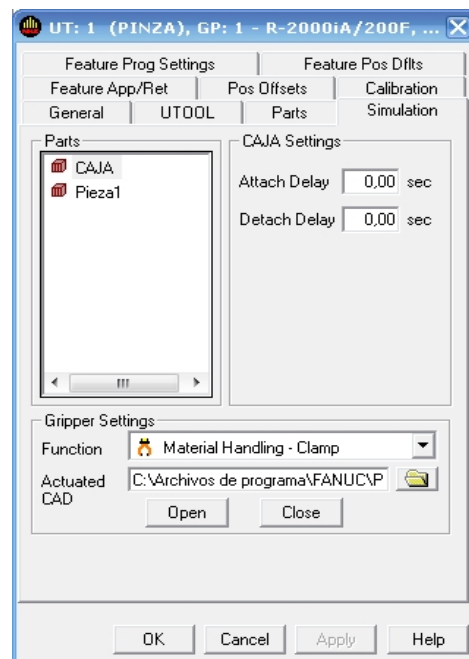


Fig. 89. Definición de características en la pestaña "Simulation".

En esta pestaña definiremos los tiempos de captura y depósito por defecto de las distintas piezas definidas. Configuraremos también la función de la herramienta utilizada (en nuestro caso *"Material Handling – Clamp"*) y asociaremos una imagen CAD "actuado", que nos permitirá animar la herramienta en el momento que cambie al estado secundario en una manipulación de alguna de las piezas o partes. En la simulación, al tratarse de una pinza, el estado primario sería la pinza abierta y el estado "actuado" correspondería con la pinza en estado cerrado con la pieza capturada.

6.4.4 Creación y definición de partes fijas (FIXTURES)

La definición de las partes fijas del sistema es primordial para la manipulación de las distintas piezas en la célula. En estas partes fijas posicionaremos los objetos que serán utilizados en el proceso.

Cada una de las piezas definidas en el apartado anterior deben de ser vinculadas a las partes fijas para que el software pueda reconocer la situación de estos objetos. En definitiva estas partes fijas serán los objetivos de la manipulación, es decir, sustentarán las piezas que deben ser capturadas y serán los soportes cuya superficie superior recibirá las piezas manipuladas.

Definiremos pues, con el siguiente procedimiento, las partes fijas de la simulación siguiendo el diseño del sistema robótico de la célula:

Accederemos al menú de configuración utilizando la ventana principal del *"Cell Browser"* y escogeremos la opción *"Add a Fixture"*, utilizando el botón derecho sobre el menú *"Fixtures"*.



Fig. 90. Creación de "fixtures" desde el menú "Cell Browser".

Seleccionaremos el cuerpo geométrico "box" y lo configuraremos según indican las imágenes que se muestran a continuación:

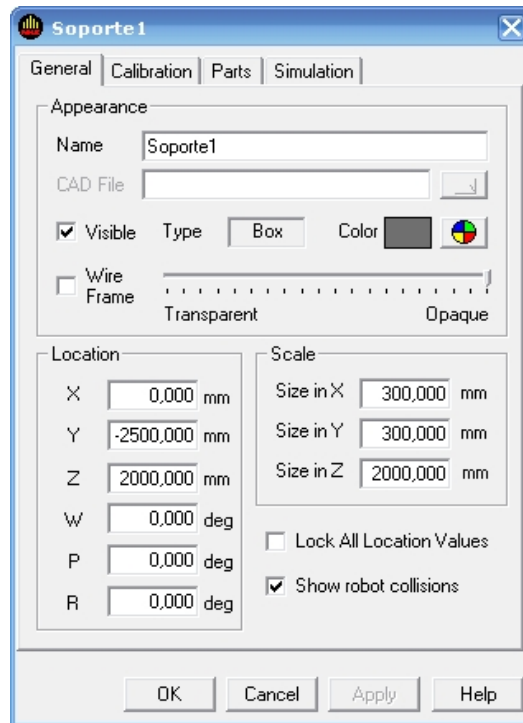


Fig. 91. Configuración de la pestaña "General" de la parte fija "Soporte1".

Accederemos a la pestaña "General" y se realizarán las siguientes acciones: nombraremos esta "fixture" como "Soporte1", la seleccionaremos como "visible", definiremos los aspectos visuales como el color, la opacidad o transparencia, la escala utilizada en el entorno de la célula y la posición en relación a la posición del brazo robótico.

Configuraremos también que, en el caso de producirse una colisión del robot con esta parte fija se indique la alarma en el periodo de "run time". Para ello utilizaremos la pestaña "Show robot collisions".

En la siguiente pestaña "*Calibration*" podemos ver todas las opciones de calibración que ofrece el software de simulación. Roboguide es capaz de generar programas de calibración de forma automática si está disponible la conexión con un robot real desde la aplicación.

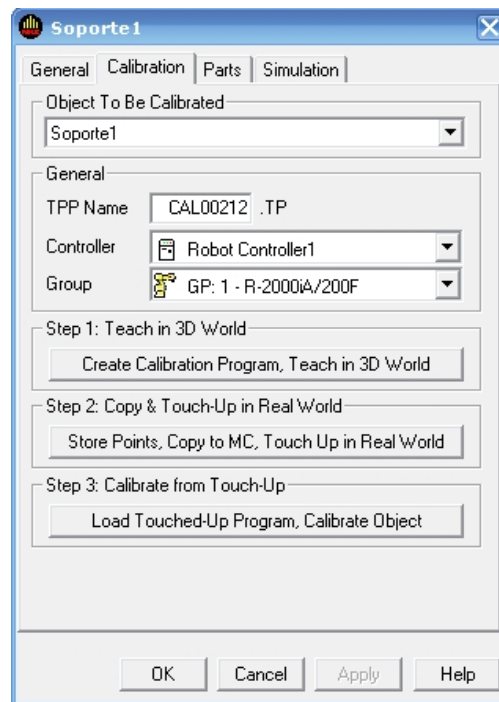


Fig. 92. Configuración de la pestaña "*Calibration*" de la parte fija "*Soporte1*".

La pestaña "*Parts*" será la que nos permita definir cada una de las partes asociadas a esta parte fija. Podremos seleccionar el objeto del conjunto de partes definidas en el sistema y posicionarlo en la parte fija tal y como se especifica en el diseño de la célula, adaptando así el futuro entorno real con el software de simulación.

Para posicionar las partes seleccionadas definiremos los valores x , y , z , w , p , r . Podremos configurar también la visibilidad de estas partes durante el "*teach time*" y el "*run time*".

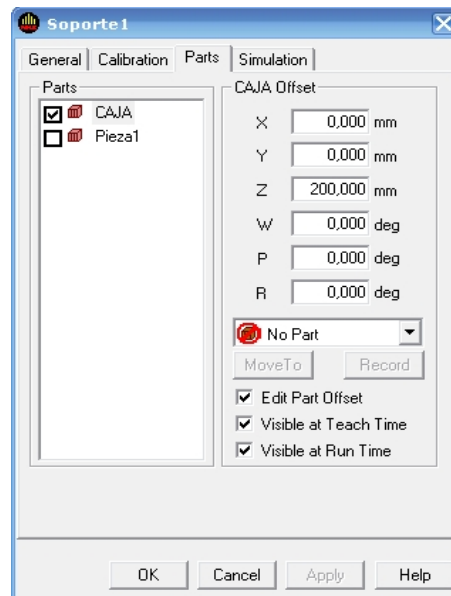


Fig. 93. Configuración de la pestaña "Parts" de la parte fija "Soporte1".

En la última pestaña "Simulation" definiremos las opciones de la parte fija en el periodo de "run time". En este menú solamente aparecerán las piezas seleccionadas en la pestaña "Parts" y podremos definir para cada una de ellas si pueden ser capturadas y/o posicionadas, además de los tiempos de retardo para estas dos acciones durante el tiempo de simulación.

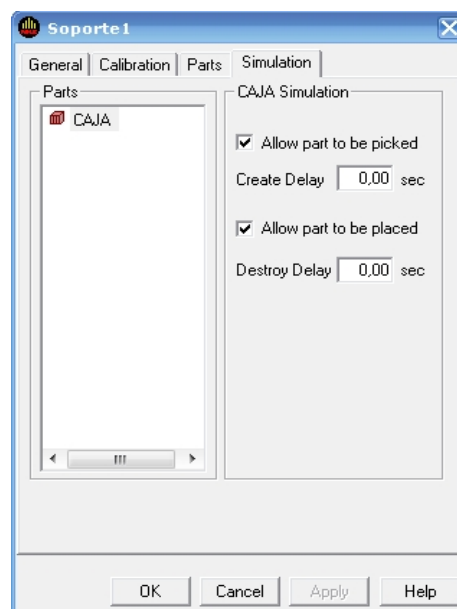
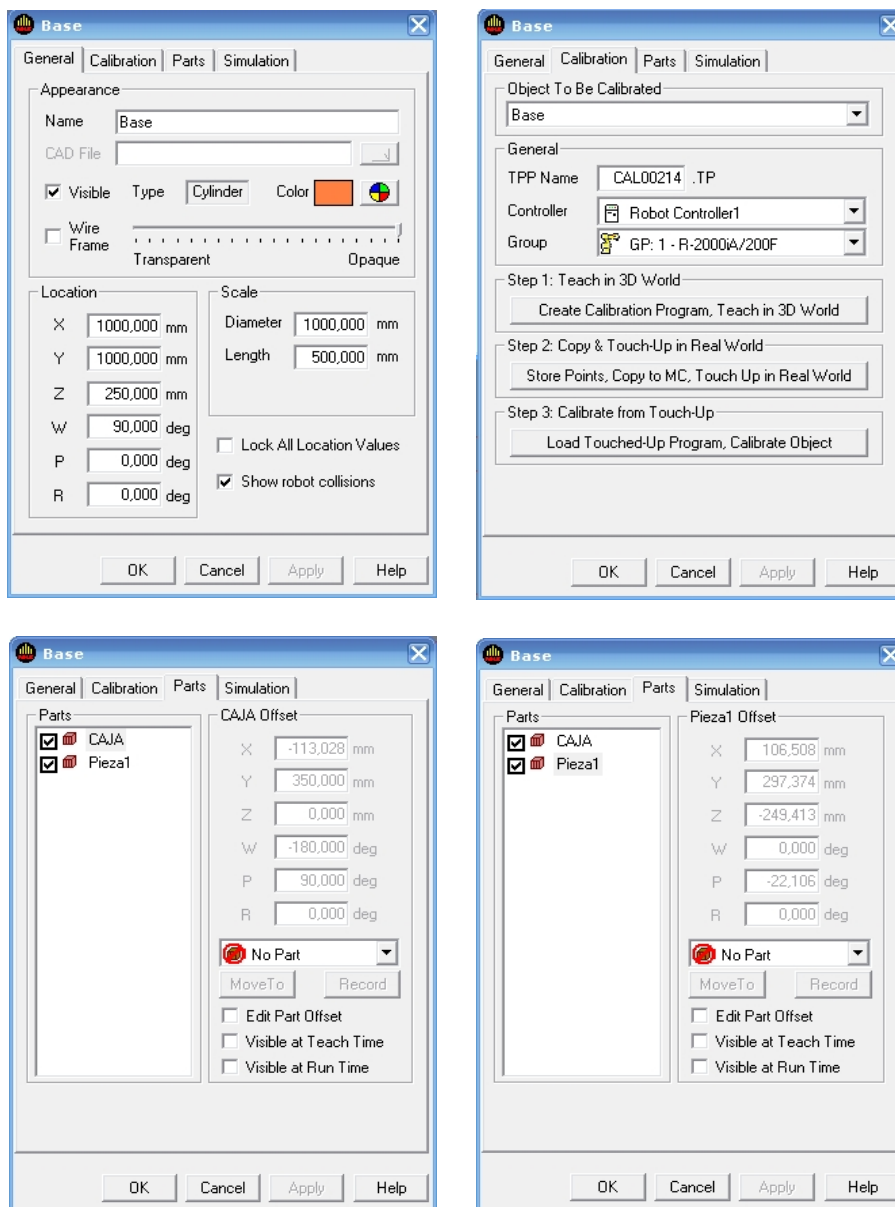


Fig. 94. Configuración de la pestaña "Simulation" de la parte fija "Soporte1".

Siguiendo el procedimiento descrito, definimos dos partes fijas más, llamadas "Soporte2" y "Soporte3", que serán iguales que el ya definido "Soporte1", salvo en altura. Definiremos distintos valores de la variable de la coordenada Z para cada uno de los soportes, siguiendo el esquema del sistema diseñado.

Además de estas partes fijas etiquetadas como "SoporteX", crearemos una parte fija de "Zona de trabajo" llamada "Base" y dos partes fijas más, iguales entre sí, que servirán como soporte para los objetos "part1".

A continuación se muestra la configuración de esta parte fija en el software de simulación, teniendo en cada una de las imágenes cada pestaña del menú de propiedades:



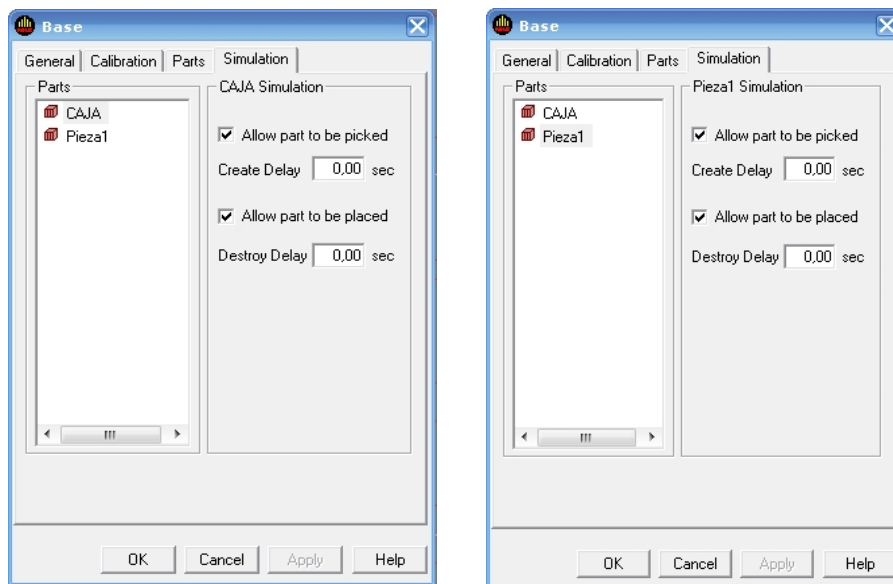
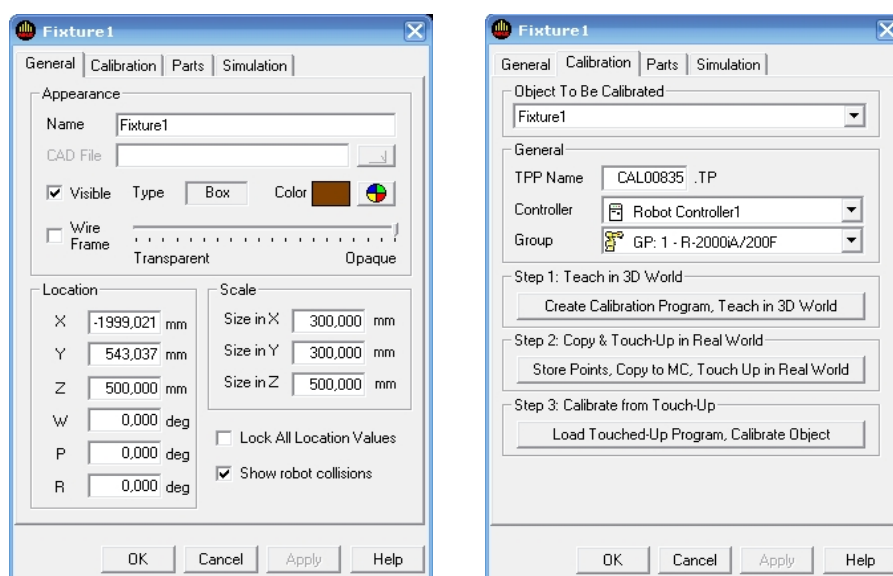


Fig. 95. Configuración de las distintas pestañas de la parte fija "Base".

Por último definimos otras dos partes fijas, que corresponderán con los soportes de los objetos "Pieza1", los cuales son iguales entre sí, salvando la posición definida de cada soporte.

La configuración de estas partes fijas será la siguiente:



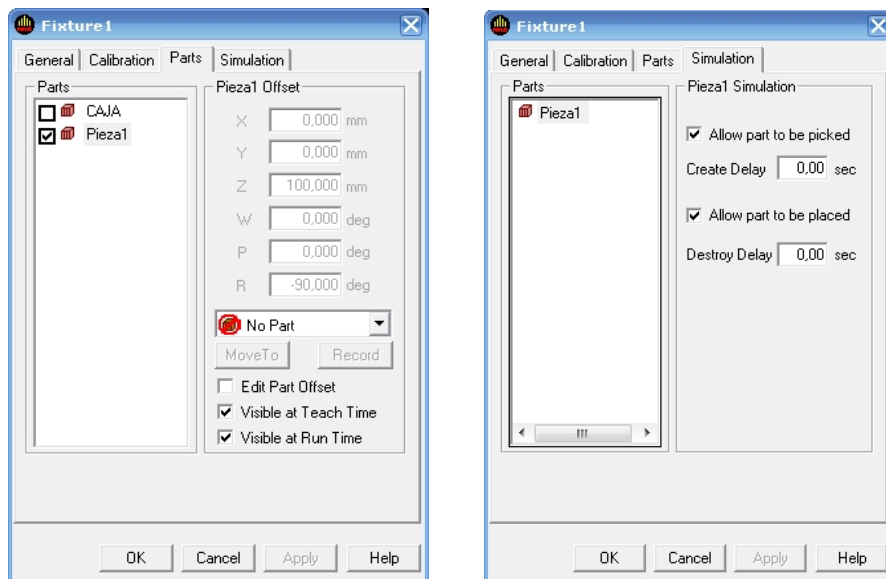


Fig. 96. Configuración de los distintas pestañas de la parte fija "Fixture1".

Una vez definidas todas las partes fijas que constarán en el sistema si revisamos el menú "Cell Browser" podremos observar que la configuración de la opción "Fixtures" queda de la siguiente manera:

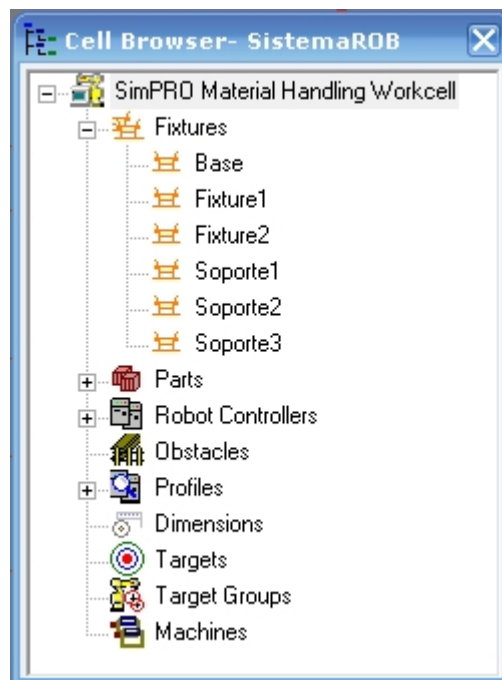


Fig. 97. Configuración de las partes fijas en el menú "Cell Browser".

5.4.5 Definición de sistema de referencia de usuario (UFRAME)

De la misma forma que en la programación por TP, en el software de simulación HandlingPRO también podemos crear sistemas de referencia definidos por el usuario "user frames", siempre y cuando las necesidades del programa lo requieran para la comodidad de la definición de puntos y ciertos movimientos (como por ejemplo una superficie inclinada definida en el entorno de trabajo del robot).

En la simulación consideramos que no es necesario definir ningún plano inclinado, ya que el sistema de referencia predefinido nos permite identificar los puntos con facilidad. El posible "UFRAME" definido en nuestro programa sería una traslación del sistema de referencia actual para que fuese relativo a una posición más cercana a un objetivo, parte u objeto del entorno de trabajo del robot. Por ello hemos definido un UFRAME, resultado de una traslación del sistema de referencia inicial, sin realizar ninguna rotación en ninguno de los ejes de coordenadas.

Este nuevo sistema de referencia "definido por el usuario" se ha posicionado en la parte fija (fixture) de la "zona de trabajo" del programa.

Para definir un nuevo UFRAME debemos acceder mediante el menú "Cell Browser" al desplegable "UserFrames". Haciendo uso del menú del botón derecho accedemos a "Uframe1 Properties" donde configuraremos el nuevo sistema de referencia de usuario.

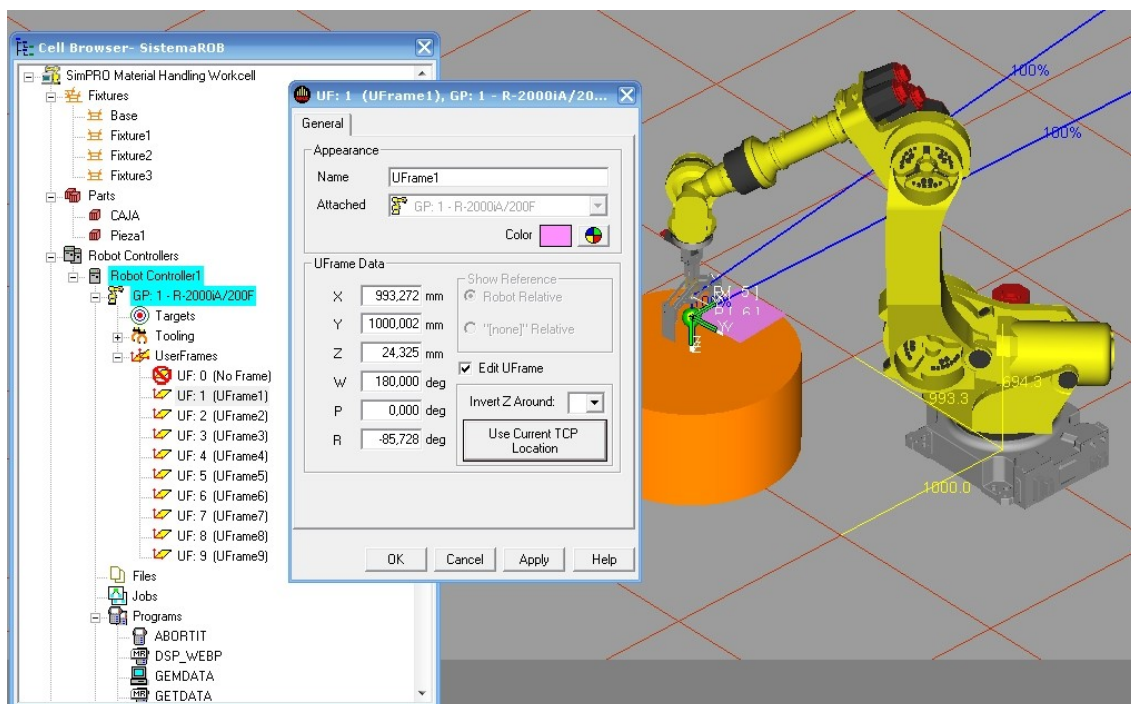


Fig. 98. Definición de UFRAME como nueva referencia.

En la ventana de configuración del *UFRAME* definiremos el nombre del nuevo sistema de referencia, el robot al que está asociado, el color y la posición del nuevo sistema de referencia. Esta posición puede ser introducida con los valores en las posiciones x, y, z y con los ángulos w, p, r, o bien utilizando el botón de "Use current TCP Location", definiendo el *UFRAME* en la posición donde se encuentra el TCP del robot en un momento determinado del entorno de la herramienta 3D.

El menú de definición de *UFRAME* también nos permite utilizar las coordenadas relativas al robot para definir el nuevo sistema de coordenadas, o no utilizar ninguna referencia, todo ello en la opción "Show Reference".

5.4.6 Creación del programa.

Después de crear cada uno de los objetos del entorno de trabajo del robot comenzamos con la implementación del programa del proceso robótico.

Para realizar la programación de los movimientos del sistema utilizaremos las herramientas que nos ofrece la aplicación, mediante el menú "Teach Program". La definición de puntos y acciones, es la base que permitirá desarrollar el algoritmo de trabajo del manipulador. Como utilidad adicional, haremos uso de la consola virtual "Teach Pendant" para poder hacer pruebas introduciendo directamente algunas coordenadas en la pestaña "Current Position" y posteriormente poder almacenar dicha posición como un nuevo punto dentro del programa.

Explicaremos pues, el método de definición de puntos para la programación del manipulador:

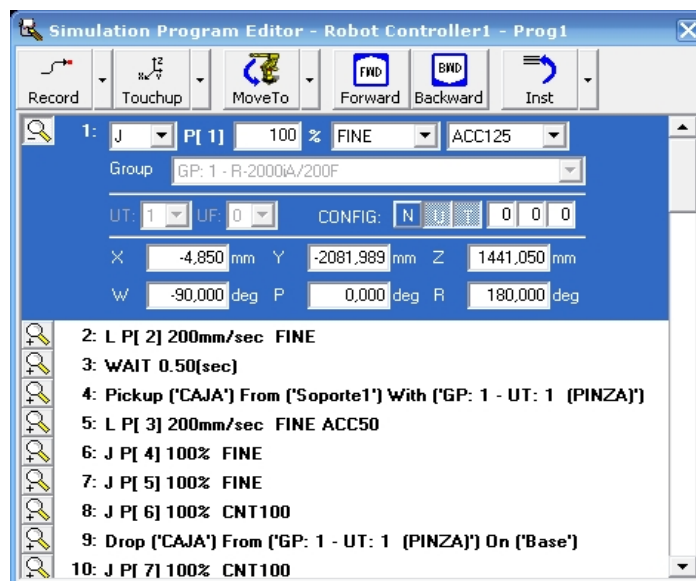


Fig. 99. Apariencia del menú "Teach Program".

Al acceder al menú de "*Teach Program*" (mediante el menú principal "*Teach*") se encuentra la interfaz para la definición de los puntos y acciones para componer las trayectorias deseadas.

En la parte superior de la ventana se encuentra la cabecera con las distintas opciones de configuración para esta definición de puntos y acciones. En la parte central e inferior observamos el espacio reservado para las instrucciones definidas, que forman el programa.

Los botones de definición de la parte superior son los siguientes:

Record: Botón utilizado para *definir* un punto en el programa con la posición actual del robot. Podemos modificar la posición del robot con la "*Teach Tool*" moviendo la esfera verde de esta herramienta (que marca el origen de referencia de la herramienta) para poder utilizar esta posición en la definición del punto. Para definir las posiciones con este botón, existe una herramienta de gran utilidad que es la barra de acceso rápido "*Move To Quick Bar*", ya que nos permite alcanzar caras, vértices, centros y aristas de los diversos objetos definidos en la célula.

También nos permite definir las coordenadas x, y, z, y los ángulos de rotación w, p, r, con respecto al sistema de referencia elegido (en nuestro caso el definido para la herramienta "Pinza"). Podemos modificar los valores de estas variables mediante la "*Teach Tool*" o generar un punto nuevo con los valores iniciados a cero, utilizando el menú desplegable de la parte derecha del botón.

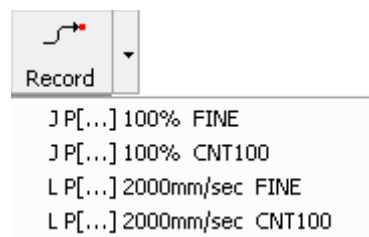


Fig. 100. Apariencia del botón "*Record*".

Touchup: Este botón nos permite *definir* un punto en el espacio pulsando posteriormente en el entorno gráfico 3D del "*teach mode*". Se comporta de forma parecida al botón "*Record*", pero no captura las variables de posición de la situación actual del robot, sino de la pulsación posterior del ratón.



Fig. 101. Apariencia del botón "*Touchup*".

MoveTo: El botón ofrece la posibilidad de *mover* el robot a cualquier punto definido en el programa del sistema, o a cualquiera de las posiciones establecidas en el menú "*Fixtures*" con definición de algún objeto ("*part*") asociado mediante el desplegable.



Fig. 102. Apariencia del botón "MoveTo".

Forward: Permite avanzar paso a paso el robot a través de los puntos definidos en el algoritmo de movimientos. Con esta herramienta podemos visualizar la secuencia de movimientos definidos con mayor comodidad.



Fig. 103. Apariencia del botón "Forward".

Backward: Permite retroceder paso a paso el robot a través de los puntos definidos en el algoritmo de movimientos. Con esta herramienta podemos visualizar la secuencia de movimientos definidos con mayor comodidad de la misma forma que con el botón "*Forward*".



Fig. 104. Apariencia del botón "Backward".

Inst: Posibilita insertar en el programa una acción sin tener que definir un punto directamente. Nos ofrece la posibilidad de capturar un objeto previamente definido en el menú "*Parts*" ("*Pickup*"); depositar un objeto definido sobre un soporte ("*Fixture*") previamente creado ("*Drop*"), insertar una instrucción de espera con tiempo definido ("*Wait*") y hacer una llamada a otro programa definido con la instrucción "*Call*".

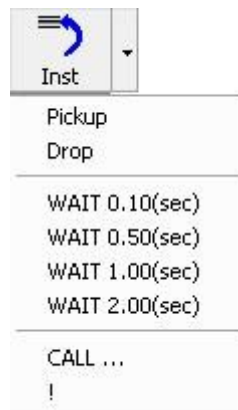


Fig. 105. Apariencia del botón "Inst".

Después de haber revisado los botones de la cabecera se explicará el formato de definición de los puntos del programa del proceso.

Las posiciones pertenecientes al "mundo" del robot se definen siguiendo el formato especificado en el apartado anterior "5.2.1 Programación TPE":

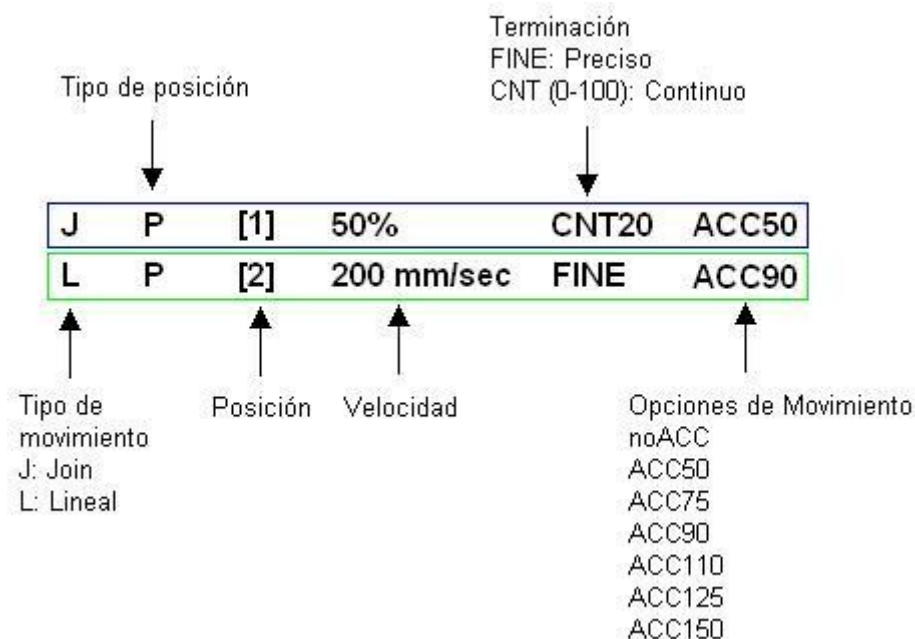


Fig. 106. Formato de instrucciones para definir posiciones.

El primer campo de la sentencia indica el tipo de movimiento, y nos permite elegir las opciones de movimiento: lineal, el cual es preciso desde el inicio hasta el final siguiendo una trayectoria delimitada; y el de enlace (join), que desplaza el TCP a cada uno de los puntos definidos *enlazando* los movimientos entre sí.

Los siguientes campos nos indican “el tipo de posición” y “la posición” del programa que estamos definiendo.

El cuarto campo determina la velocidad con la que se va a realizar el movimiento. Dependiendo de si el movimiento es lineal (L) o de enlace (J) podremos determinar la velocidad de forma más precisa (en mm/seg) o de forma más genérica (en %) respectivamente.

El quinto campo indica el tipo de finalización del proceso. Éste puede ser continuo (CNT) o preciso (FINE).

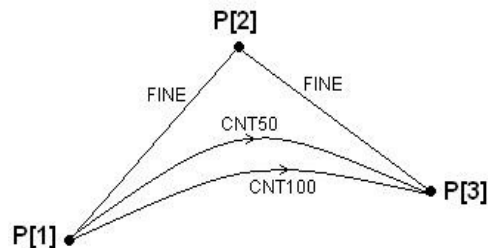


Fig. 107. Tipos de finalización de movimiento.

El último campo es opcional y determina la dinámica del movimiento. En el caso del menú “Teach Program” podemos escoger la opción ACC, que determina el régimen de aceleración/deceleración al moverse de un punto a otro.

Para definir una acción en el menú, como por ejemplo capturar un objeto o depositarlo, debemos actual de la siguiente forma:

Pickup (CAJA)
From (Soporte1)
With (GP: 1 - UT: 1 (PINZA))

Fig. 108. Definición de acción “Pickup”.

Drop (CAJA)
From (GP: 1 - UT: 1 (PINZA))
On (Base)

Fig. 109. Definición de acción “Drop”.

Para determinar una acción de manipulación deberemos definir en primer lugar el objeto creado en el menú "Parts" que queremos manipular, la herramienta específica del menú "Tooling" con la que queremos trabajar y la parte fija determinada en "Fixtures" donde queremos realizar la acción.

Para crear un punto, usaremos como herramienta el menú del "Teach Pendant" virtual "Current Position", ya que nos permite mover el brazo robótico a un punto del espacio del robot, siempre que esté permitido en su rango de movimiento.

Esta opción nos permite hacer pruebas antes de grabar un punto y visualizar que la posición deseada es posible, y consecuentemente que el robot se posicione para alcanzar el punto que estamos definiendo.

El menú del "Teach Pendant" virtual es el que se muestra a continuación:

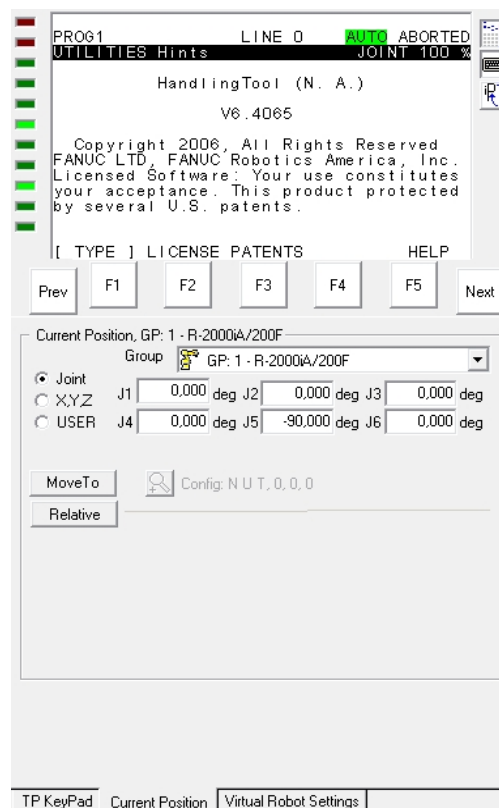


Fig. 110. Apariencia del menú "Current Position" del TPE.

Este menú ofrece tres formas de introducción de datos para alcanzar una posición concreta con el TCP. Primero, podemos especificar los grados de cada uno de los ejes (*Joint*); segundo, las coordenadas XYZ y rotaciones WPR del punto a alcanzar respecto al sistema de referencia base; o tercero, las coordenadas XYZ y rotaciones WPR en un sistema de referencia de usuario (*UFRAME*).

Habiendo analizado las herramientas para la definición de posiciones y creación del algoritmo del proceso, se va a describir a continuación la definición de algunos puntos de ejemplo en referencia a aproximaciones a algunos objetos de la célula de trabajo.

- Aproximación para depositar "CAJA" en "Soporte1:"

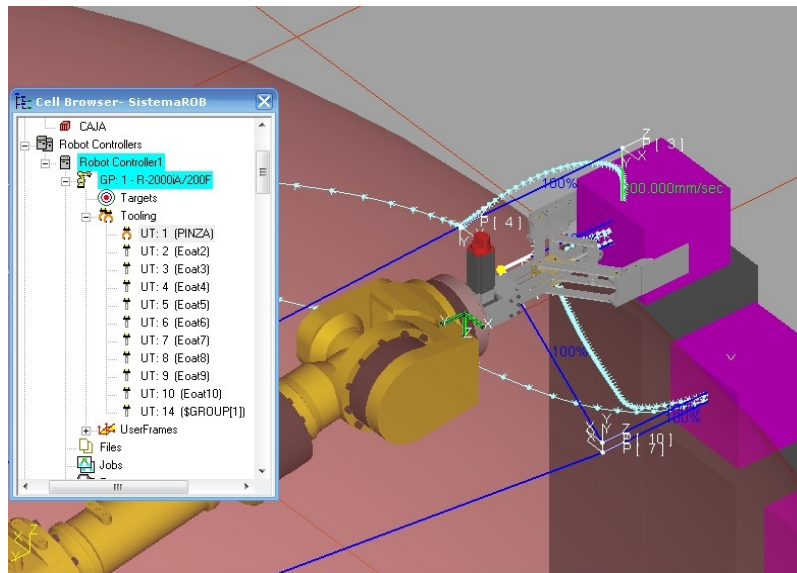


Fig. 111. Definición de posición para la aproximación al "Soporte1".

- Aproximación para capturar "CAJA" de "Soporte3"

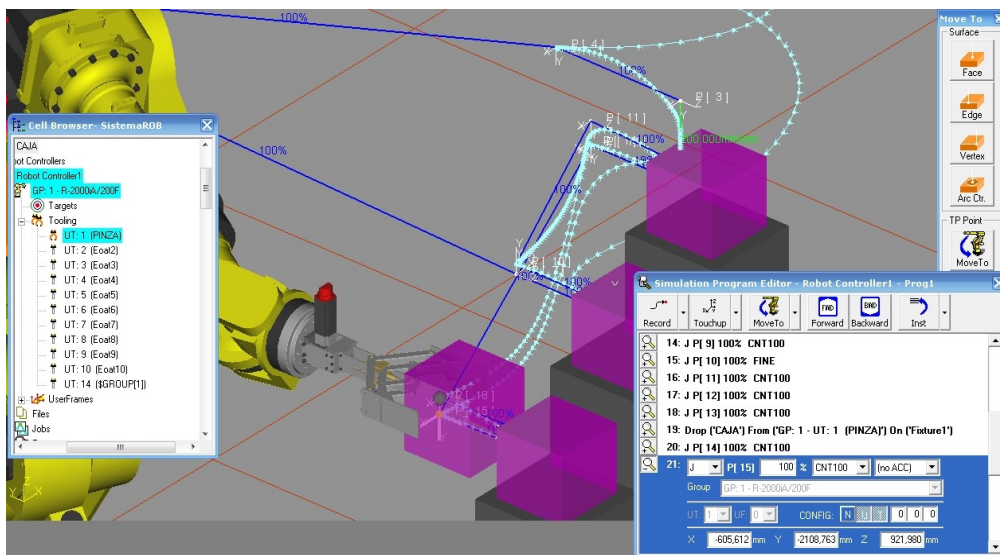


Fig. 112. Definición de posición para la aproximación al "Soporte3".

- Posicionado para depositar "CAJA" en "Soporte2"

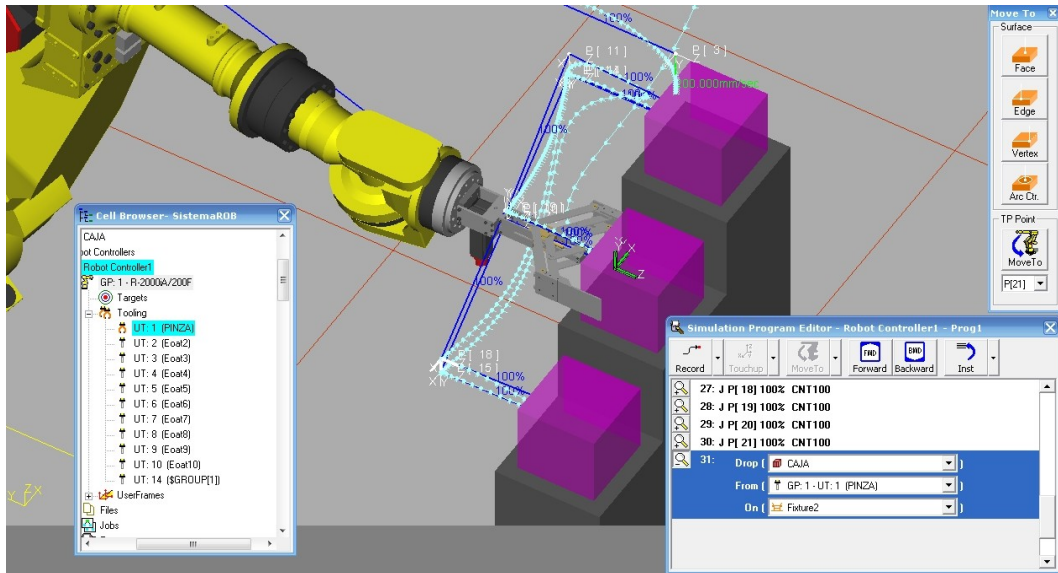


Fig. 113. Definición de posición para depositar "CAJA" en "Soporte2".

- Aproximación para capturar "Pieza1" en "Fixture2"

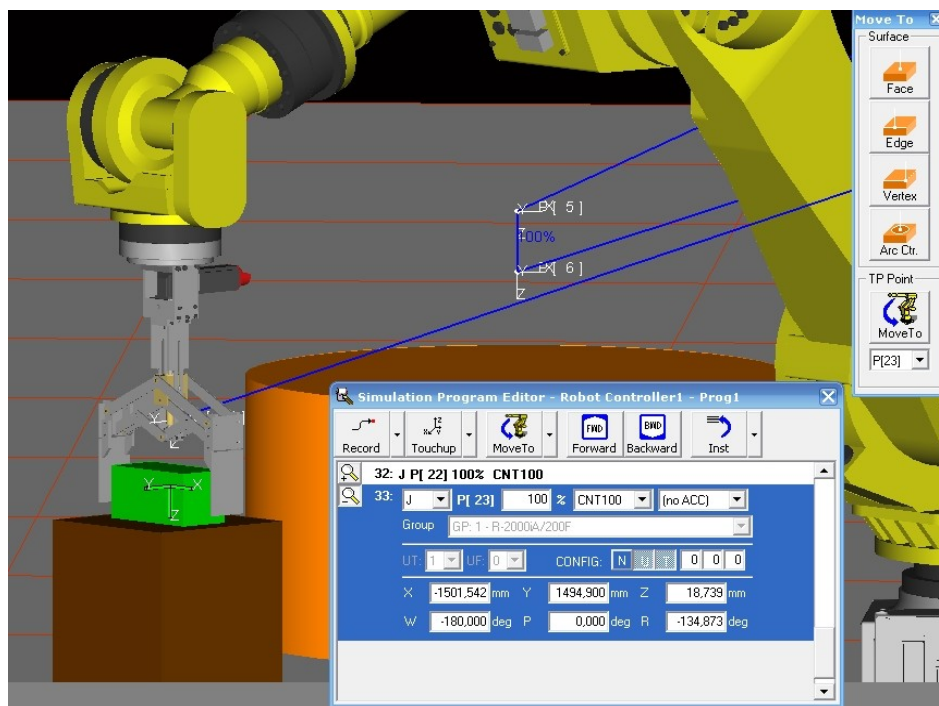


Fig. 114. Definición de posición para capturar "Pieza1" en "Fixture2".

- Separación de "Base" después de depositar "Pieza1"

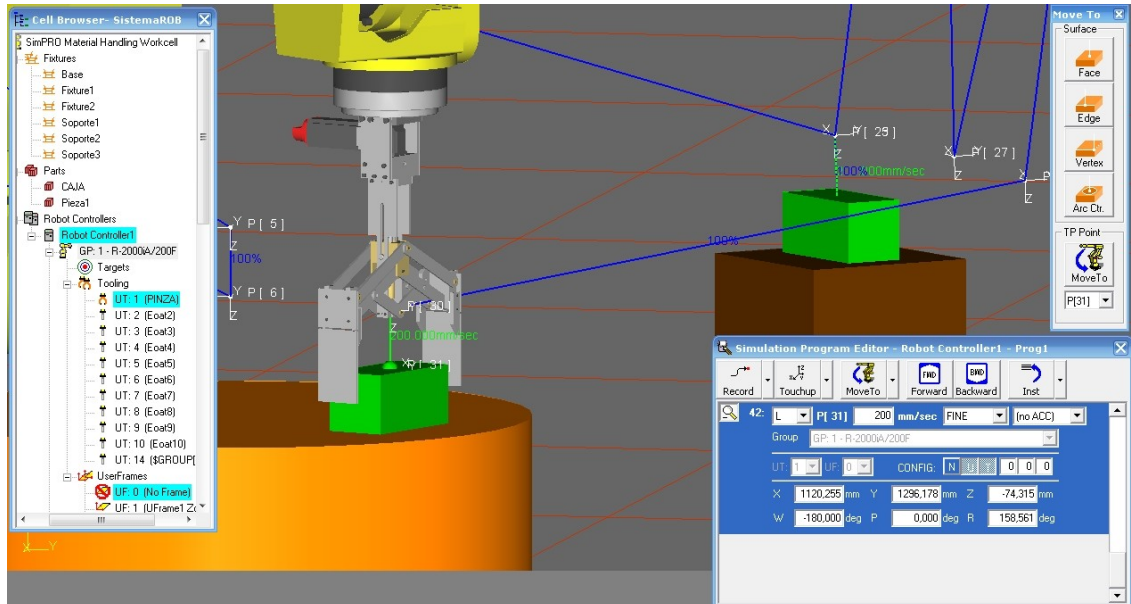


Fig. 115. Definición de posición para la separación de base al depositar "Pieza1".

- Aproximación a "Fixture1" para capturar "Pieza1".

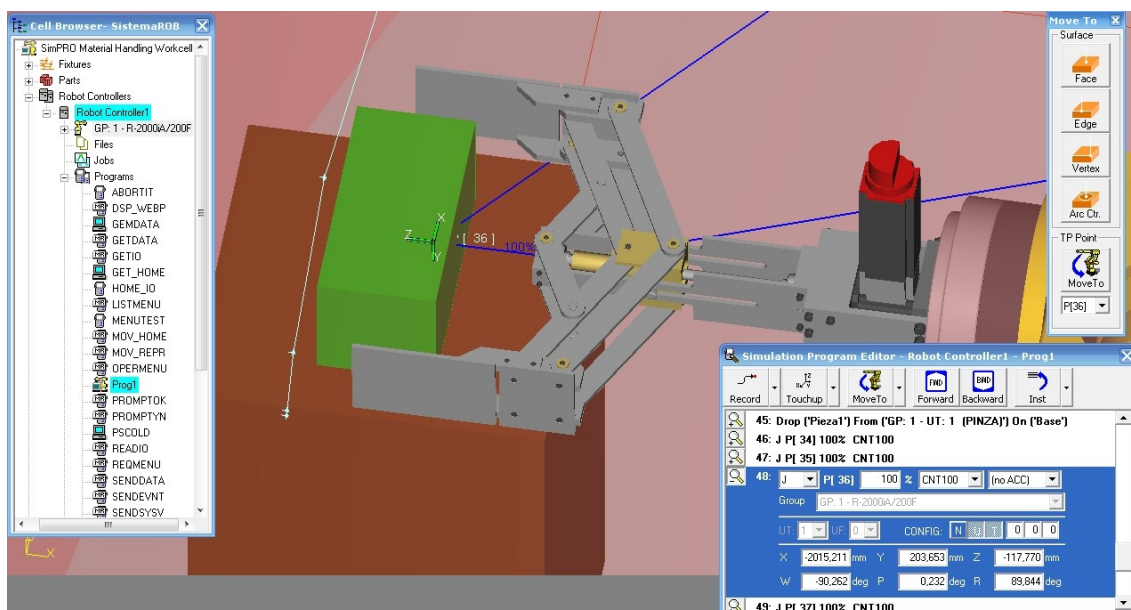


Fig. 116. Definición de posición para capturar "Pieza1" de "Fixture1".

5.4.7 Simulación del proceso.

Con el sistema robótico totalmente realizado y completado el algoritmo de definición de posiciones, ejecutamos la simulación del proceso para supervisar el sistema programado y cada una de las acciones previstas que se realizarán en la célula robótica.

Mediante los botones de la barra de acceso rápido, o a través del menú "Run Panel", ubicado en el menú principal "Teach", podremos ejecutar la simulación para visualizar el proceso completo, obteniendo información de las posibles colisiones, movimientos erróneos, o imprecisiones en aproximaciones de los movimientos del brazo. De esta manera identificaremos los errores de una forma visual y los corregiremos fácilmente antes de que aparezcan en el sistema real de producción, ya que el efecto proactivo al trabajar de forma virtual, es una de las principales ventajas de los sistemas de simulación.

El menú "Run Panel" permite controlar varios aspectos de la simulación, para poder ajustarla según las necesidades que afecten a la supervisión del sistema.

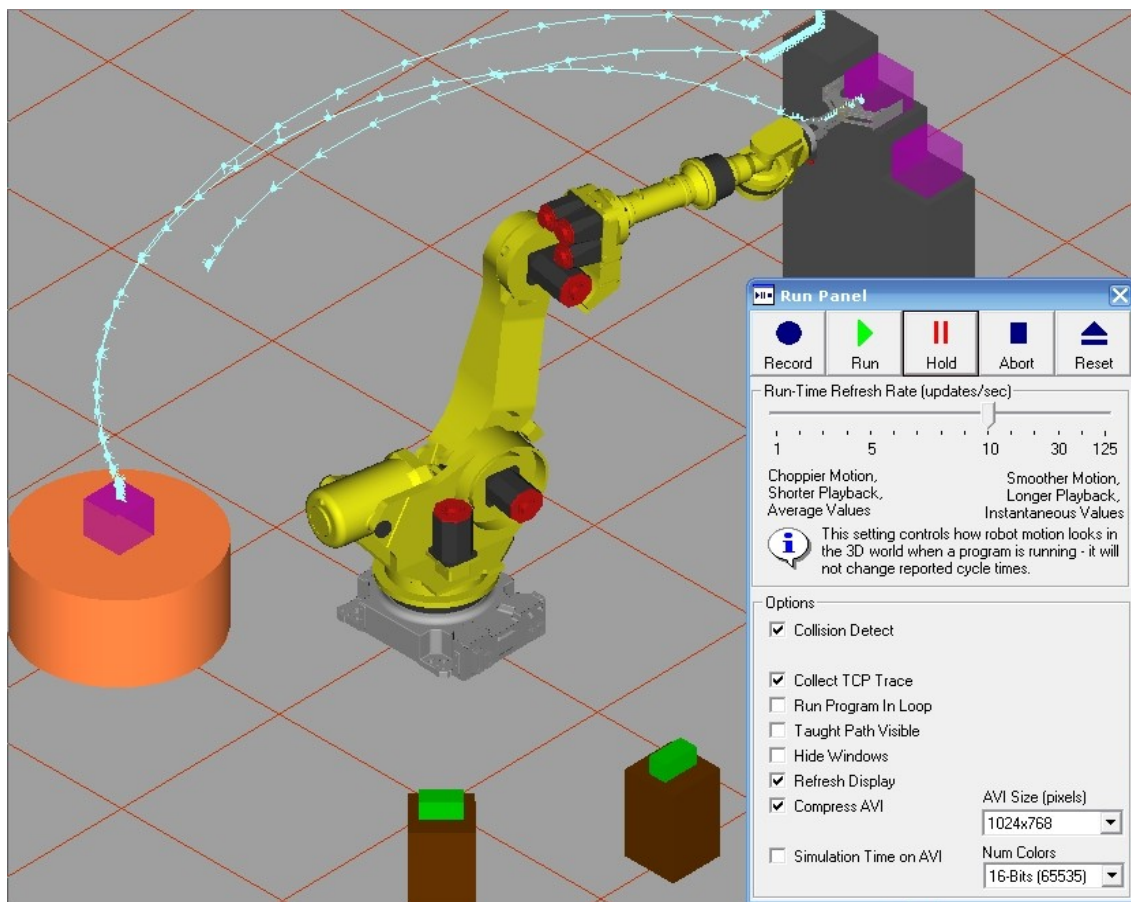


Fig. 117. Utilización del menú "Run Panel" para la simulación.

El menú *"Run Panel"* consta de los botones de ejecución en la parte superior, y varias opciones de configuración en la parte central e inferior de la ventana.

En la parte *"Run time refresh rate"* podemos configurar el número de *"frames"* o imágenes por segundo que va a mostrar la simulación del proceso, configurando así la fluidez de la animación de la simulación.

En *"Options"* podemos activar o desactivar algunas opciones concretas, que detallamos a continuación:

- Detección de colisiones.
- Definición de la traza del TCP (trayectorias del brazo).
- Ejecutar en bucle (modo *"loop"*).
- Trazado de puntos definidos en el programa.
- Esconder ventanas en simulación.
- Actualizar pantalla.
- Comprimir archivo de video AVI.

Una vez visualizada la simulación, corregiremos las colisiones que se produzcan, los errores y los desajustes detectados en el proceso. Redefiniremos los puntos *"problemáticos"* para conseguir el proceso con las posiciones y la precisión deseada. De esta forma ajustaremos todas las trayectorias del proceso robótico del sistema industrial para obtener el programa definitivo.

La aplicación Roboguide permite realizar la Calibración del sistema automáticamente si tenemos conectado el robot a la aplicación de simulación. Sería el momento de realizar la calibración y cargar el programa en el entorno real (*"upload"*) mediante la exportación del programa.

Finalmente, tendríamos la célula robótica y lista para entrar en producción.

6. Ampliación de la célula de trabajo.

Con la célula robótica programada y montada en producción, podrá darse el caso de que sea necesario ampliar el sistema, debido a necesidades que puedan surgir en el proceso industrial. Debemos entonces rediseñar la célula, aunque ésta se encuentre en plena producción y con todos los sistemas a pleno rendimiento.

Es por ello que, para implementar la ampliación del sistema robótico, reutilizaremos el diseño y la programación realizada en la herramienta de simulación Roboguide® HandlingPRO® con anterioridad.

La reprogramación de la célula de trabajo mediante el software nos permitirá rediseñar el nuevo sistema sin tener que intervenir el entorno real, el cual se encuentra funcionando en producción. De esta forma reduciremos al máximo el impacto en la cadena de trabajo de la planta industrial.

Después de haber completado el diseño y la implementación del proceso, supervisaremos el nuevo sistema como ya se realizó con la primera célula; corregiremos los fallos, desajustes y colisiones que puedan aparecer, asegurando así la funcionalidad total del nuevo sistema en el entorno virtual.

Para realizar la ampliación de la célula, se va a configurar un segundo manipulador, además de añadir nuevas partes fijas que intervengan en el sistema y una nueva parte a manipular, además de los objetos ya definidos en el proceso anterior.

Se pretende controlar los movimientos de los dos manipuladores para que puedan trabajar ambos de forma simultánea, y sin que se produzcan colisiones entre sí, completando el proceso de una forma eficiente y controlando los espacios de trabajo de ambos robots (ajustando con especial cuidado la zona de trabajo común, que será la zona donde puedan presentarse mayores problemas).

A continuación se presentará el procedimiento llevado para realizar la ampliación de la célula de trabajo ya definida en los apartados anteriores.

6.1 Rediseño de la célula robótica.

Afrontamos el rediseño partiendo de la célula de trabajo inicial, ajustando las nuevas necesidades de trabajo en el sistema y componiendo una nueva célula que resuelva los mencionados requerimientos.

Se añade un manipulador adicional "Robot2" al sistema existente, dos nuevas zonas de trabajo, y se eliminan dos de los tres soportes de cajas, quedando en diseño de la célula como se muestra la figura siguiente:

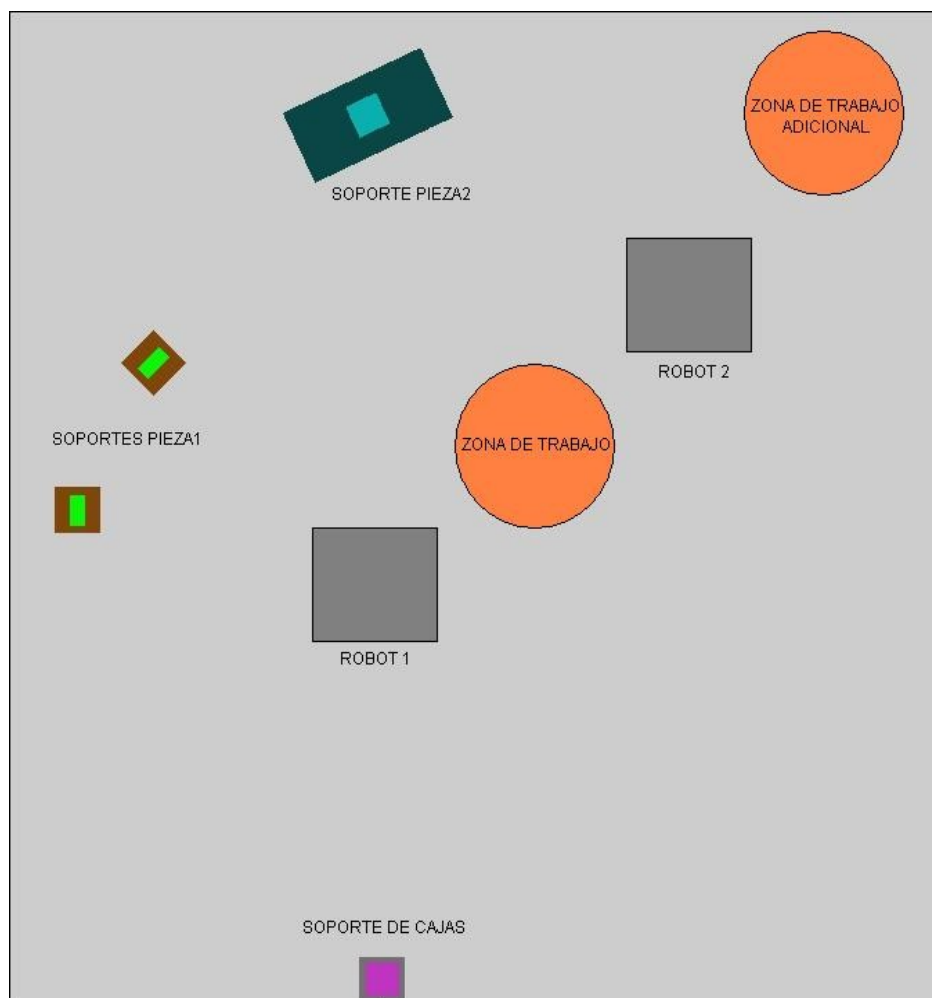


Fig. 118. Ampliación y rediseño de la célula de trabajo.

Las nuevas zonas son:

- *La zona del manipulador adicional.* En este espacio se sitúa el nuevo robot "Robot2" que se integra a la célula de trabajo.
- *La zona de soporte de la Pieza 2.* Esta zona proveerá una nueva pieza al sistema, la cual intervendrá en las nuevas zonas definidas.
- *La zona de trabajo adicional.* Está destinada a una nueva acción mecanizada. Se trasladarán piezas para ser manipuladas y tratadas, para posteriormente ser devueltas a la zona de trabajo principal.

El nuevo diagrama de estados se compondrá de seis acciones posibles y de seis posiciones permitidas. La ejecución de estas acciones y el posicionamiento en estos puntos se realizará de forma simultánea entre los dos manipuladores, sincronizando los movimientos para evitar toda colisión que pueda producirse en el entorno de la célula de trabajo.

Las acciones del nuevo proceso son:

- Captura de "Caja"
- Captura de "Pieza1"
- *Captura de "Pieza2"*
- Depósito de "Caja"
- Depósito de "Pieza1"
- *Depósito de "Pieza2"*

Las posiciones permitidas en el entorno del manipulador son:

- Soporte 1
- Base (Zona de trabajo)
- Fixture 1
- Fixture 2
- *Soporte A*
- *Base A*

El nuevo proceso constará de los siguientes pasos, especificados para cada manipulador:

<i>Robot1:</i>	<i>Robot2:</i>
<ul style="list-style-type: none"> – Posición inicial.* – Posicionamiento en "Soporte1". – Captura de "Caja". – Posicionamiento en "Base". – Depósito de "Caja" – Posicionamiento en "Fixture2" – Captura de "Pieza1" – Posicionamiento en "Base". – Depósito de "Pieza1" – Posicionamiento en "Fixture1". – Captura de "Pieza1" – Posicionamiento en "Fixture2" – Depósito de "Pieza1". – Posición de espera.* – Posicionamiento en "Base". – Captura de "Caja". – Posicionamiento en "Soporte1". – Depósito de "Caja". – Posicionamiento en "Base". – Captura de "Pieza1". – Posicionamiento en "Fixture1". – Depósito de "Pieza1". – Posición final.* 	<ul style="list-style-type: none"> – Posición inicial.* – Posicionamiento en "SoporteA". – Captura de "Pieza2". – Posicionamiento en "Base". – Depósito de "Pieza2" – Posición de espera. – Posicionamiento en "Base" – Captura de "CAJA" – Posicionamiento en "BaseA". – Depósito de "CAJA" – Posición de espera. – Posicionamiento en "BaseA" – Captura de "CAJA" – Posicionamiento en "Base". – Depósito de "CAJA" – Captura de "Pieza1" – Posicionamiento en "BaseA". – Depósito de "Pieza1" – Posición de espera. – Posicionamiento en "BaseA" – Captura de "Pieza1" – Posicionamiento en "Base". – Depósito de "Pieza1" – Captura de "Pieza2" – Posicionamiento en "SoporteA" – Depósito de "Pieza2". – Posición final.*

Gracias a las secuencias de ejecución de los dos manipuladores, mediante el análisis de acciones y de posiciones del sistema, podemos generar los nuevos gráficos de red de Petri (separados para cada manipulador para facilitar la lectura).

Cabe destacar de la introducción de un nuevo estado de espera, utilizado por el "Robot2" en algunos puntos del proceso para evitar colisiones con el "Robot1".

Robot 1:

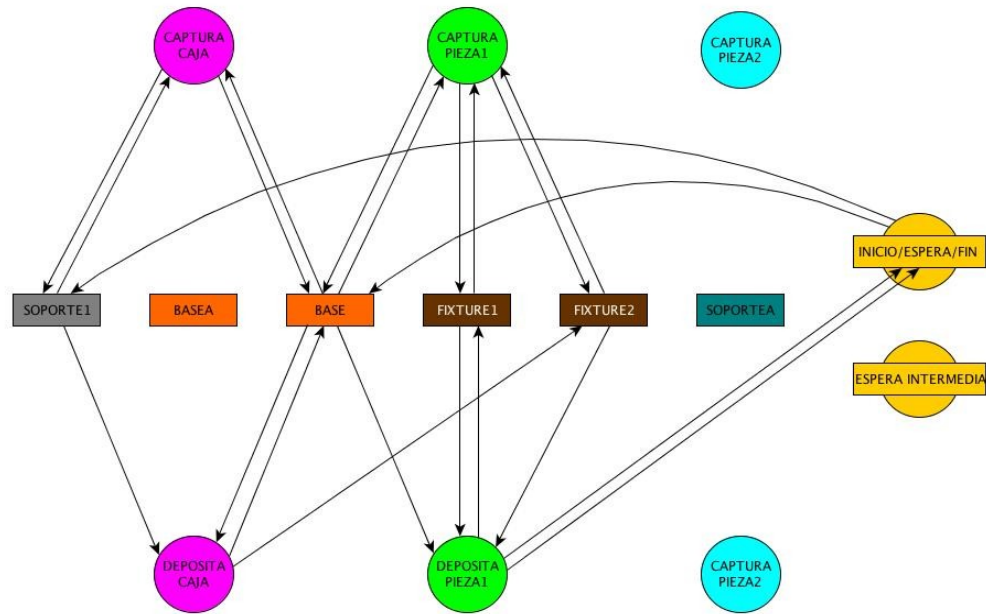


Fig. 119. Red de Petri para la secuencia del "Robot1".

Robot 2:

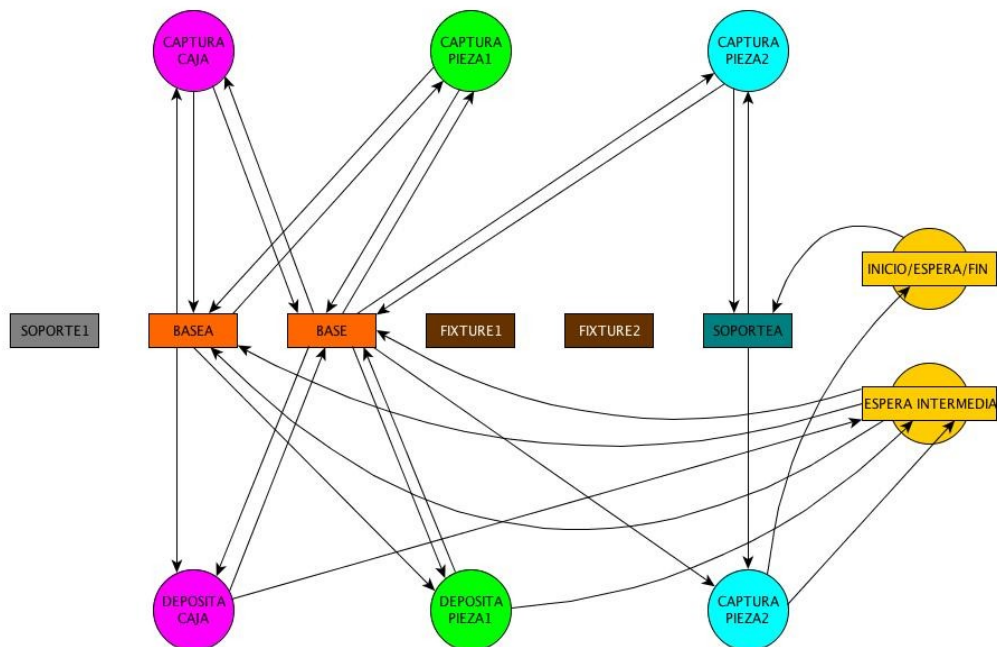


Fig. 120. Red de Petri para la secuencia del "Robot2".

6.2 Configuración de la célula con robot adicional.

Para configurar la nueva célula, en primer lugar debemos añadir un nuevo manipulador al entorno de trabajo, utilizando el menú *"Cell Browser"* y sobre el menú *"Robot Controllers"*, utilizando el botón derecho del ratón, escogeremos la opción *"Add Robot"*. Seguidamente aparecerá el asistente de creación de robot, ya utilizado en la primera creación del *"Robot1"*, donde debemos escoger la opción *"Create a copy of an existing robot"*. De esta forma obtendremos un robot igual que el anterior para poder comenzar a configurar la nueva célula.

En primer lugar debemos configurar la posición de este nuevo robot, ya que nos aparecerá en la misma posición que el primero (solapados).

Después debemos de volver a definir la herramienta *UTOOL* utilizada para este segundo robot. Utilizamos la herramienta *"Pinza"* ya definida anteriormente.

Por último, crearemos el resto de objetos del entorno de la nueva célula.

Creación del objeto *"Pieza2"*:

En el menú *"Cell Browser"* sobre el botón *"Parts"*, utilizamos el botón derecho del ratón para escoger la opción *"Add Part"*, configurándolo como se indica a continuación:

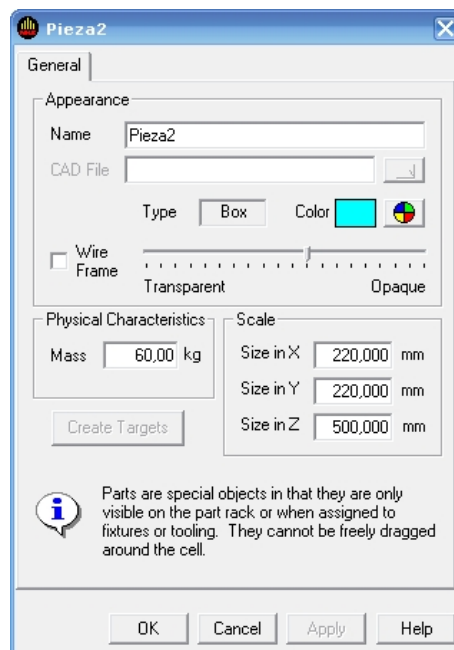


Fig. 121. Menú de propiedades del objeto *"Pieza2"*.

Creación de soporte "Pieza2":

En el menú "Cell Browser" sobre el botón "Fixtures", utilizamos el botón derecho del ratón para escoger la opción "Add Fixture", configurándolo como se indica a continuación:

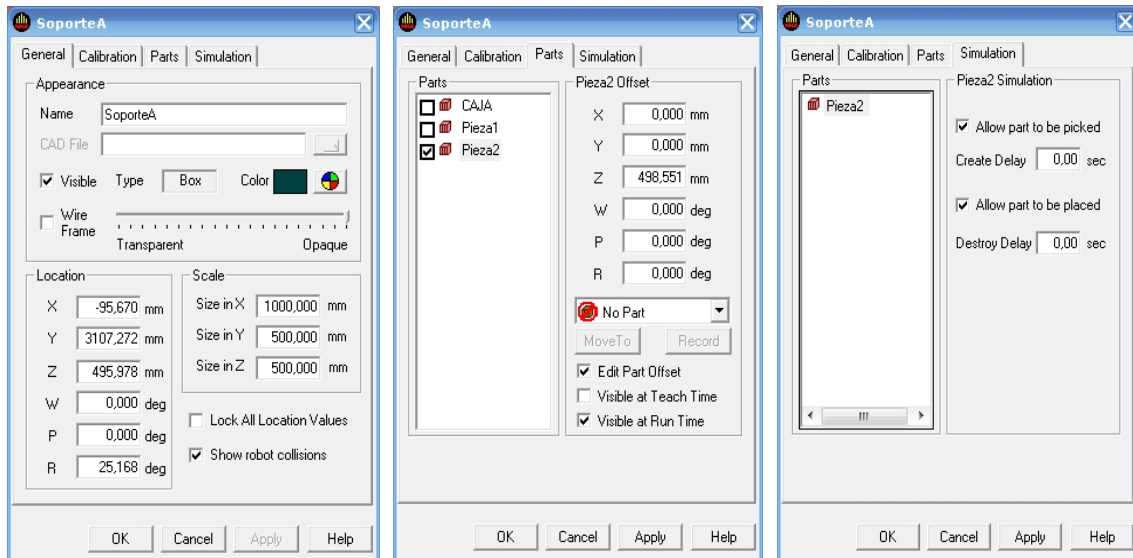


Fig. 122. Menús de propiedades de la parte fija "SoporteA".

Creación de la zona de trabajo adicional "BaseA":

Seleccionando la parte fija "Base" en el menú "Cell Browser", o seleccionando el objeto "Base" en el entorno gráfico 3D, podemos escoger la opción "Copy Base" para posteriormente utilizar "Paste Base" y reconfigurarla según las necesidades requeridas.

Debemos configurar la posición de la nueva parte fija, además de la etiqueta, para comenzar a configurar el resto de opciones.

Configuraremos este nuevo soporte según se indica en las imágenes siguientes:

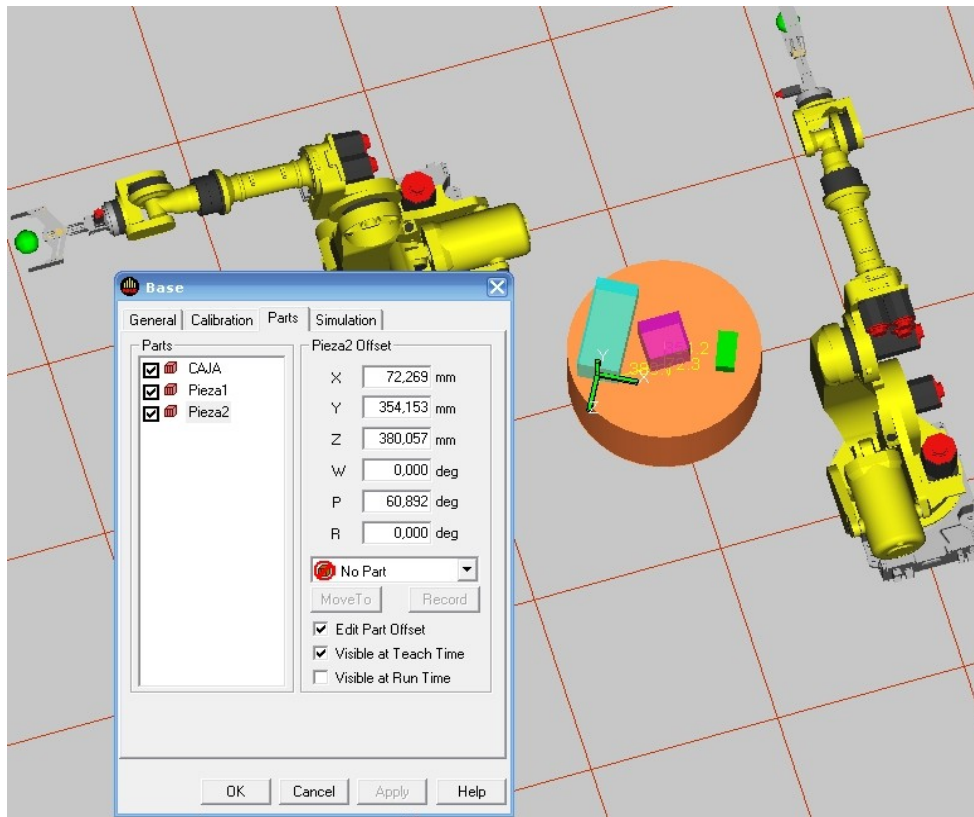


Fig. 123. Configuración de objetos vinculados a la parte fija "BaseA".

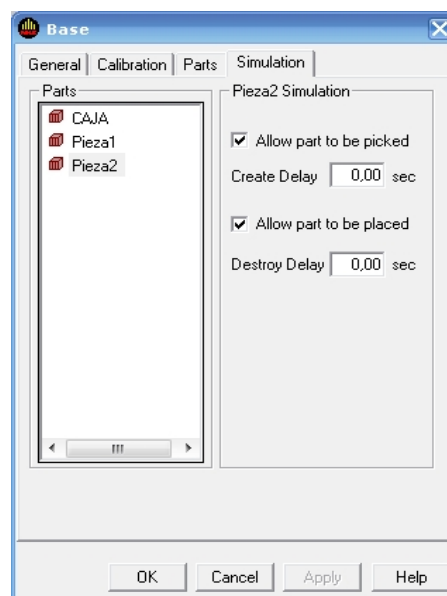


Fig. 124. Configuración de las opciones de simulación de la parte fija "BaseA".

6.3 Creación del programa ampliado.

Siguiendo la definición de la secuencia de acciones y movimientos expuesta en el apartado 6.1, procedemos a implementar cada una de las posiciones y cada una de las interacciones con los objetos manipulados por los dos robots.

Antes de comenzar a implementar los algoritmos de manipuladores, debemos hacer especial hincapié en la posibilidad de colisiones entre ambos y controlar esta situación en todo momento. Gracias a la herramienta de simulación y su detección de colisiones podremos redefinir en todo momento los programas y tener esta situación de colisión en la célula bajo control

En la siguiente imagen se muestra la zona de colisión de los brazos del sistema, remarcada con una línea de color negro:

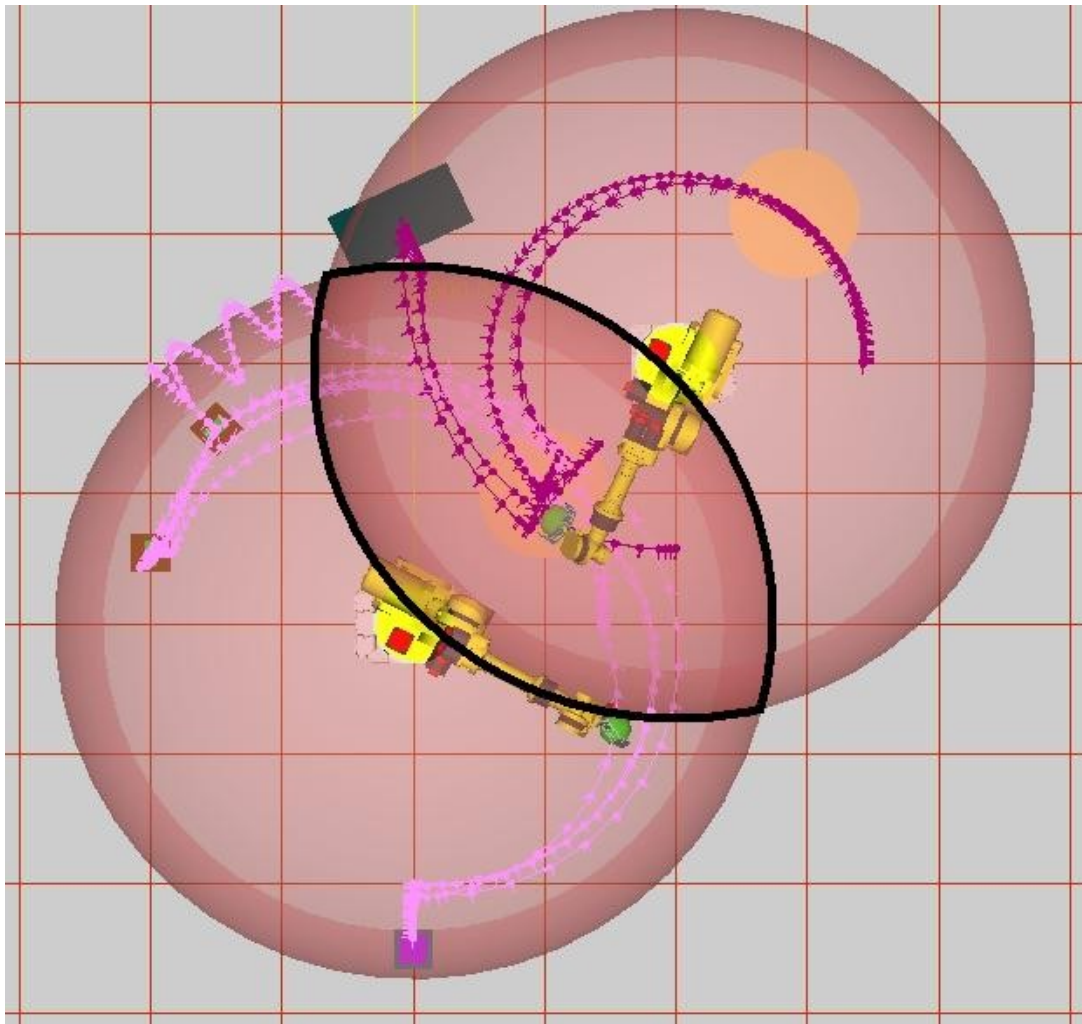


Fig. 125. Zona de colisión entre los manipuladores del sistema.

Teniendo siempre en cuenta la posibilidad de colisión, definiremos entonces las posiciones y acciones implementando los algoritmos de ambos robots. A continuación se muestran algunos puntos definidos en la herramienta de simulación, ilustrados con las capturas del entorno 3D:

- Aproximación a "SoporteA" para capturar "Pieza2":

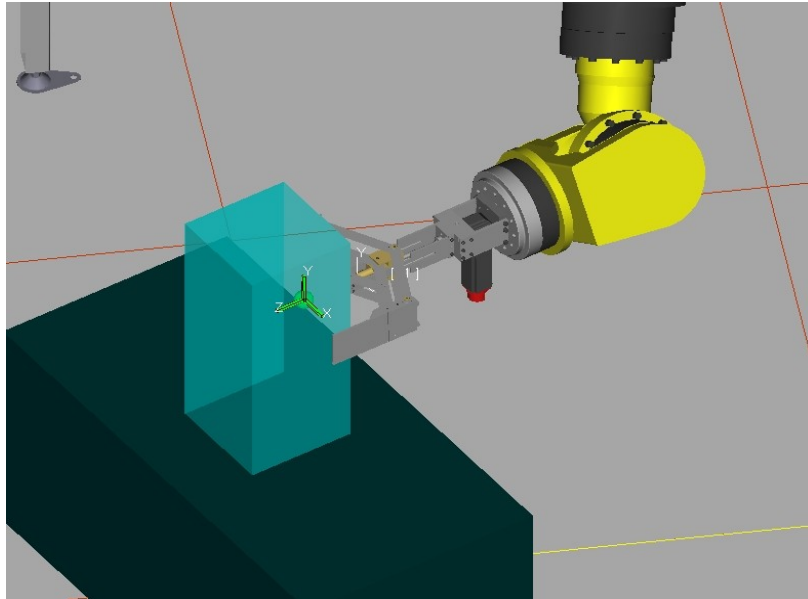


Fig. 126. Definición de posición para la aproximación al "SoporteA".

- Aproximación a "BaseA" para capturar "CAJA":

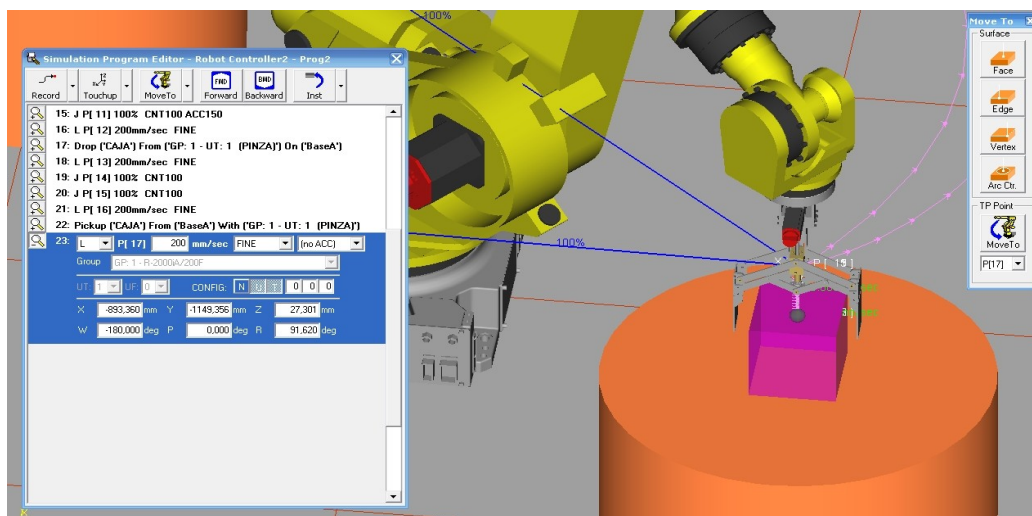


Fig. 127. Definición de posición para la aproximación a la "BaseA".

- Depósito de "Pieza1" en "Base":

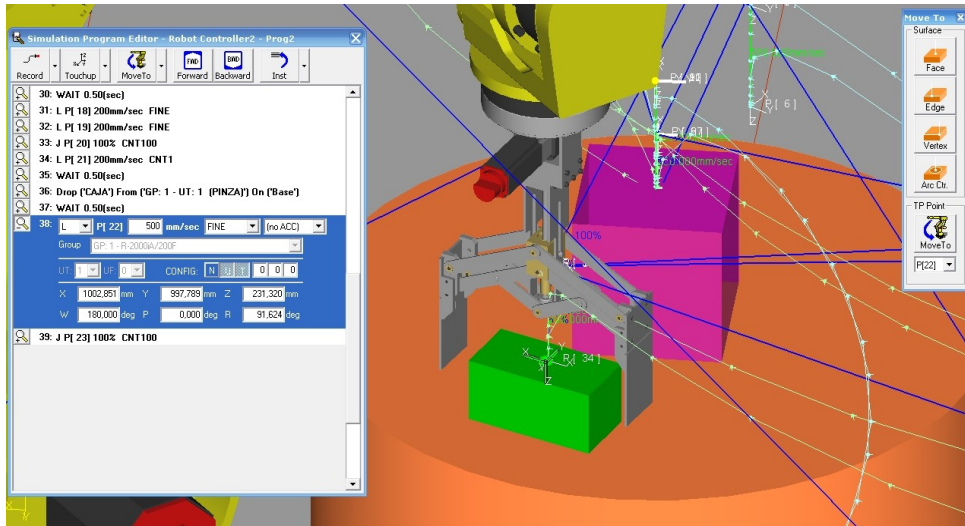


Fig. 128. Definición de posición para "Pieza1" en "Base".

Cuando tengamos los programas implementados de los dos manipuladores procederemos a simular las trayectorias y comprobar si se producen colisiones, para posteriormente corregir los puntos definidos erróneos, aproximaciones a partes fijas, velocidades de movimiento y tiempos de espera en la ejecución del proceso.

La siguiente imagen muestra la implementación del sistema con las definiciones de los programas de los dos robots ("Prog1" y "Prog2").

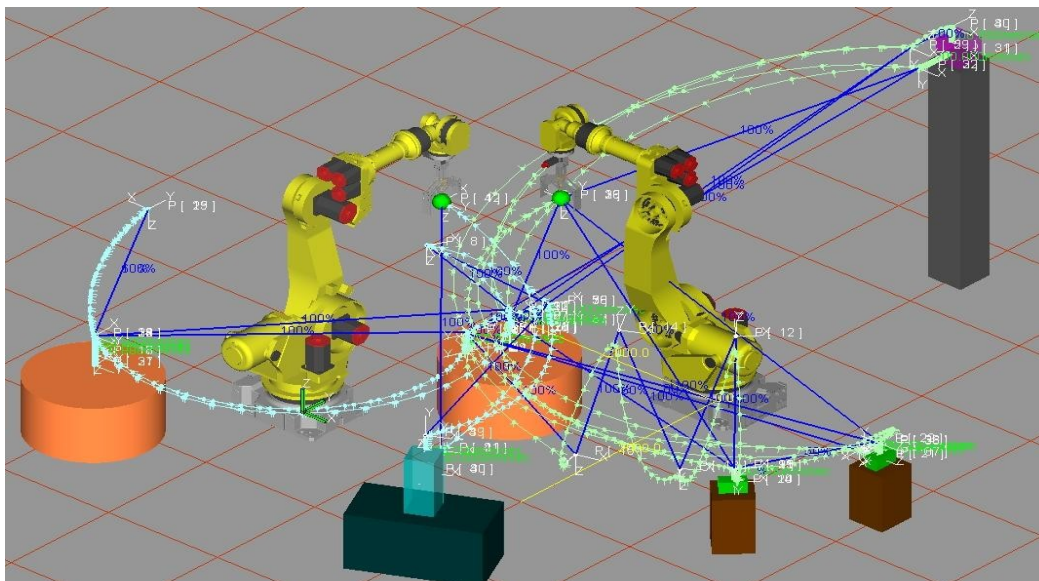


Fig. 129. Imagen de la ampliación del sistema, implementado en "Prog1" y "Prog2".

6.4 Simulación del sistema.

Con el nuevo sistema robótico totalmente realizado, ejecutamos la simulación del proceso para supervisar el sistema programado y comprobamos las posibles colisiones, supervisando la zona crítica donde pueden producirse estas "interferencias" entre los manipuladores.

Después de numerosas pruebas y de diversos ajustes en el proceso, se completa la secuencia de comandos sin obtener ningún choque entre los brazos robóticos.

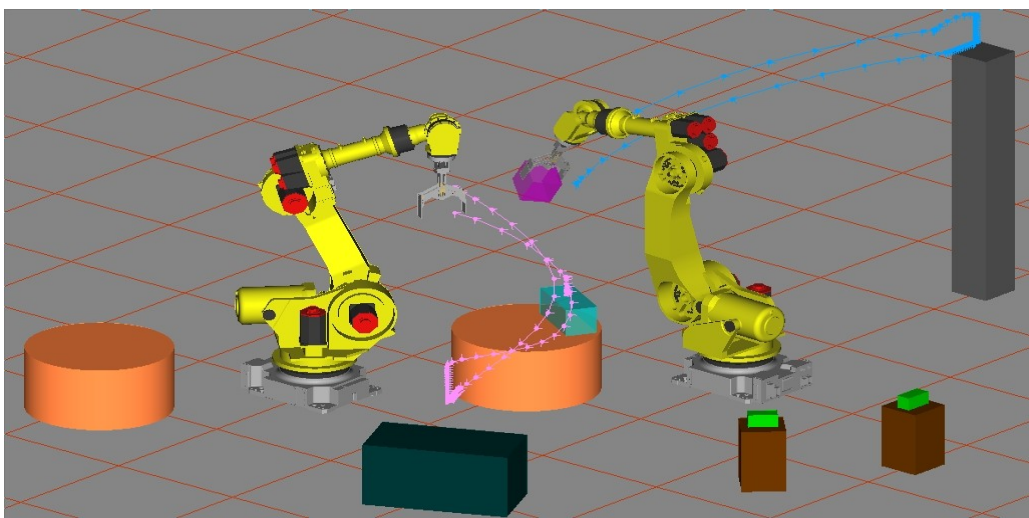


Fig. 130. Imagen de la secuencia de simulación de la célula de trabajo (1).

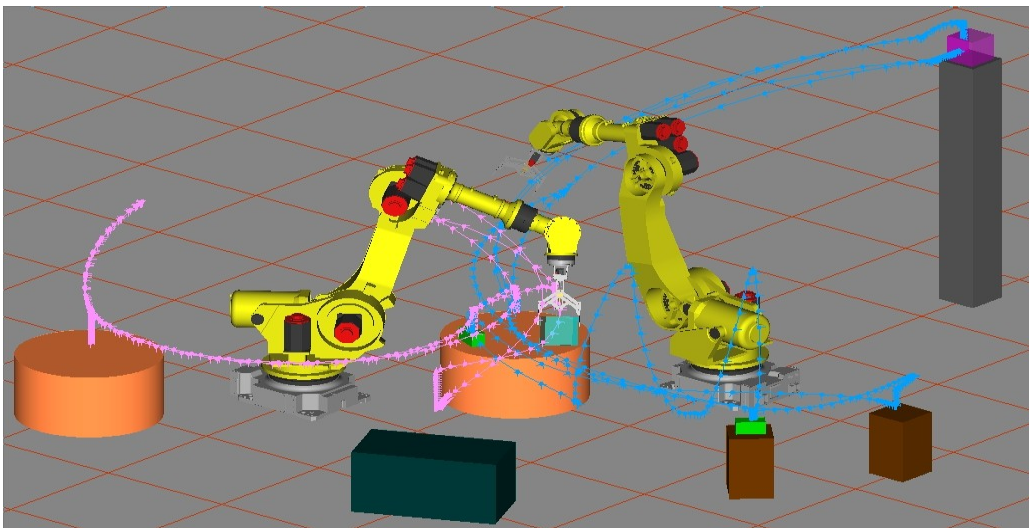


Fig. 131. Imagen de la secuencia de simulación de la célula de trabajo (2).

7. Evaluación de resultados y conclusiones.

Gracias a la utilización de herramientas de simulación hemos conseguido analizar un proceso robótico cualquiera, de forma virtual, antes de tener desplegado todo el sistema real. Hemos simulado un proceso tipo de "pick and place", característico del sector industrial, comprobando su funcionamiento, supervisando las trayectorias de movimiento, evaluando el proceso completo de forma global y en particular hemos analizado y verificado cada una de las acciones puntuales del sistema.

Por tanto, podemos decir que hemos cumplido los objetivos propuestos al inicio, ya que hemos conseguido *determinar y corregir* de forma *eficiente* las incidencias que podrían haber surgido en una implementación realizada directamente sobre un entorno real.

Se ha analizado también el concepto de ingeniería simultánea, y se ha evaluado su aplicación a los entornos de simulación industrial. Hemos podido comprobar que se adapta perfectamente a la forma de trabajo de este concepto de ingeniería.

Se han introducido los aspectos matemáticos de los movimientos de traslación y rotación de un sistema de referencia, para poder comprender los fundamentos de los movimientos del manipulador en la célula de trabajo.

Para el desarrollo de este trabajo final de carrera, se ha aprendido y analizado el funcionamiento de la aplicación de simulación Roboguide® de FANUC®, una de las herramientas más utilizadas en entornos reales para realizar la programación "offline" de células robóticas industriales. Se ha escogido un robot industrial, utilizado en entornos productivos, con un alto rendimiento y una amplia versatilidad en cadenas de producción robotizadas, gracias a sus grados de libertad, capacidad de carga útil y alcance de manipulación.

Se ha programado una célula de trabajo, definiendo cada una de las posiciones y cada una de las acciones llevadas a cabo por los manipuladores. Para ello se han usado las herramientas disponibles en el interfaz visual que ofrece la aplicación de simulación (*teach mode*), para posteriormente simular todo el proceso mediante la misma herramienta (*run mode*).

Con la simulación del proceso, hemos concretado los errores de posición en el sistema, mejorado la precisión de las aproximaciones a los objetos, detectando las colisiones producidas con los objetos, partes fijas y otros robots del entorno. Estos errores detectados se han corregido en la propia herramienta de simulación, para posteriormente realizar más pruebas y ajustar el sistema hasta conseguir el rendimiento requerido.

Se ha rediseñado la célula de trabajo representando una ampliación del sistema, añadiendo un segundo manipulador además de nuevos objetos. Se ha analizado y controlado la posibilidad de colisión entre los manipuladores en la ampliación de la célula y se ha aprendido a dominar esta situación, implementando los programas de ambos sin que se produzcan colisiones.

Las posibles ampliaciones de este sistema serían: la integración con un sistema de visión para el posicionamiento automático, la integración con sensores para realizar tareas mecanizadas dependientes de los registros de entrada y salida y la aplicación de una herramienta de simulación similar a un robot para poder implementar y realizar la calibración del sistema con la posición real.

8. Bibliografía.

"Curso de programación TPE" - Fanuc Robotics.

"Concurrent Engineering" - John R. Hartley.

"Apuntes de robótica para la asignatura" - Juan Domingo.

"Automatización Integrada" - Revista de robótica, Pulsar.

Handling Tool Operators Manual – Fanuc Robotics

Roboguide Introduction – Fanuc Robotics

Concepto de ingeniería simultánea y modelado por simulación. - Josep Ribó Pablo.

Morfología de los robots industriales. - Antonio Romeo.

Menú de ayuda de la aplicación Roboguide

www.fanucrobotics.com

9. Anexos.

Programa del sistema inicial:

```
1: J P[ 1] 100% FINE ACC125
2: L P[ 2] 200mm/sec FINE
3: WAIT 0.50(sec)
4: Pickup ('CAJA') From ('Soporte1') With ('GP: 1 - UT: 1 (PINZA)')
5: WAIT 0.50(sec)
6: L P[ 3] 200mm/sec FINE ACC50
7: J P[ 4] 100% FINE
8: J P[ 5] 100% FINE
9: L P[ 6] 200mm/sec FINE
10: WAIT 0.50(sec)
11: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('Base')
12: WAIT 0.50(sec)
13: J P[ 7] 100% CNT100
14: J P[ 8] 100% CNT100
15: L P[ 9] 200mm/sec FINE
16: WAIT 0.50(sec)
17: Pickup ('CAJA') From ('Soporte2') With ('GP: 1 - UT: 1 (PINZA)')
18: WAIT 0.50(sec)
19: L P[ 10] 200mm/sec FINE
20: L P[ 11] 200mm/sec FINE
21: J P[ 12] 100% CNT100
22: L P[ 13] 300mm/sec CNT100
23: L P[ 14] 200mm/sec FINE
24: WAIT 0.50(sec)
25: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('Soporte1')
26: WAIT 0.50(sec)
27: L P[ 15] 400mm/sec CNT100
```

28: J P[16] 100% CNT100
29: L P[17] 200mm/sec FINE
30: WAIT 0.50(sec)
31: Pickup ('CAJA') From ('Soporte3') With ('GP: 1 - UT: 1 (PINZA)')
32: WAIT 0.50(sec)
33: L P[18] 200mm/sec FINE
34: J P[19] 100% CNT100
35: J P[20] 100% CNT100
36: L P[21] 200mm/sec FINE
37: L P[22] 200mm/sec FINE
38: WAIT 0.50(sec)
39: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('Soporte2')
40: WAIT 0.50(sec)
41: L P[23] 400mm/sec CNT100
42: J P[24] 100% CNT100
43: L P[25] 200mm/sec FINE
44: WAIT 0.50(sec)
45: Pickup ('Pieza1') From ('Fixture2') With ('GP: 1 - UT: 1 (PINZA)')
46: WAIT 0.50(sec)
47: J P[26] 100% CNT100
48: J P[27] 100% CNT100
49: J P[28] 100% CNT100
50: J P[29] 100% CNT100
51: J P[30] 100% CNT100
52: J P[31] 100% CNT100
53: J P[32] 100% CNT100
54: L P[33] 200mm/sec FINE
55: WAIT 0.50(sec)

56: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('Base')
57: WAIT 0.50(sec)
58: J P[34] 100% CNT100
59: J P[35] 100% CNT100
60: L P[36] 200mm/sec FINE
61: WAIT 0.50(sec)
62: Pickup ('Pieza1') From ('Fixture1') With ('GP: 1 - UT: 1 (PINZA)')
63: WAIT 0.50(sec)
64: L P[37] 200mm/sec FINE
65: J P[38] 100% CNT100
66: L P[39] 200mm/sec FINE
67: WAIT 0.50(sec)
68: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('Fixture2')
69: WAIT 0.50(sec)
70: L P[40] 400mm/sec CNT100
71: J P[41] 100% CNT100
72: WAIT 2.00(sec)
73: J P[42] 100% CNT100
74: L P[43] 200mm/sec FINE
75: WAIT 0.50(sec)
76: Pickup ('CAJA') From ('Base') With ('GP: 1 - UT: 1 (PINZA)')
77: WAIT 0.50(sec)
78: J P[44] 100% CNT100
79: L P[45] 200mm/sec FINE
80: L P[46] 200mm/sec FINE
81: WAIT 0.50(sec)
82: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('Soporte3')
83: WAIT 0.50(sec)

84: L P[47] 200mm/sec FINE
85: L P[48] 200mm/sec CNT100
86: J P[49] 100% CNT100
87: L P[50] 200mm/sec FINE
88: WAIT 0.50(sec)
89: Pickup ('Pieza1') From ('Base') With ('GP: 1 - UT: 1 (PINZA)')
90: WAIT 0.50(sec)
91: J P[51] 100% CNT100
92: J P[52] 100% CNT100
93: L P[53] 200mm/sec CNT100
94: WAIT 0.50(sec)
95: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('Fixture1')
96: WAIT 0.50(sec)
97: L P[54] 200mm/sec FINE
98: J P[55] 100% CNT100

Programas de la ampliación del sistema:

Prog1 (Robot1):

```
1: J P[ 1] 100% CNT100
2: J P[ 2] 100% CNT100
3: L P[ 3] 200mm/sec FINE
4: WAIT 0.50(sec)
5: Pickup ('CAJA') From ('Soporte1') With ('GP: 1 - UT: 1 (PINZA)')
6: WAIT 0.50(sec)
7: L P[ 4] 200mm/sec FINE ACC50
8: J P[ 5] 100% FINE
9: J P[ 6] 100% FINE
10: L P[ 7] 200mm/sec FINE
11: WAIT 0.50(sec)
12: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('Base')
13: WAIT 0.50(sec)
14: J P[ 8] 100% CNT100
15: WAIT 0.50(sec)
16: J P[ 9] 60% CNT100
17: L P[ 10] 200mm/sec FINE
18: WAIT 1.00(sec)
19: Pickup ('Pieza1') From ('Fixture2') With ('GP: 1 - UT: 1 (PINZA)')
20: WAIT 1.00(sec)
21: J P[ 11] 100% CNT100
22: J P[ 12] 100% CNT100
23: J P[ 13] 100% CNT100
24: J P[ 14] 100% CNT100
25: J P[ 15] 100% CNT100
26: J P[ 16] 100% CNT100
27: J P[ 17] 100% CNT100
```


28: L P[18] 200mm/sec FINE
29: WAIT 0.50(sec)
30: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('Base')
31: WAIT 0.50(sec)
32: L P[19] 400mm/sec FINE
33: J P[20] 80% CNT100
34: L P[21] 200mm/sec FINE
35: WAIT 0.50(sec)
36: Pickup ('Pieza1') From ('Fixture1') With ('GP: 1 - UT: 1 (PINZA)')
37: WAIT 0.50(sec)
38: L P[22] 200mm/sec FINE
39: J P[23] 50% CNT100
40: L P[24] 200mm/sec FINE
41: WAIT 0.50(sec)
42: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('Fixture2')
43: WAIT 0.50(sec)
44: L P[25] 400mm/sec CNT100
45: J P[26] 50% CNT100
46: WAIT 3.00(sec)
47: J P[27] 100% CNT100
48: J P[28] 50% CNT50
49: WAIT 0.50(sec)
50: Pickup ('CAJA') From ('Base') With ('GP: 1 - UT: 1 (PINZA)')
51: WAIT 0.50(sec)
52: J P[29] 100% CNT100
53: J P[30] 100% CNT100
54: L P[31] 200mm/sec FINE
55: WAIT 1.00(sec)

56: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('Soporte1')
57: WAIT 1.00(sec)
58: L P[32] 200mm/sec FINE
59: J P[33] 100% CNT100
60: L P[34] 200mm/sec FINE
61: WAIT 0.50(sec)
62: Pickup ('Pieza1') From ('Base') With ('GP: 1 - UT: 1 (PINZA)')
63: WAIT 0.50(sec)
64: J P[35] 100% FINE
65: J P[36] 100% CNT100
66: L P[37] 200mm/sec CNT100
67: WAIT 0.50(sec)
68: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('Fixture1')
69: WAIT 0.50(sec)
70: L P[38] 200mm/sec FINE
71: J P[39] 100% CNT100

Prog2 (Robot2):

```
1: J P[ 1] 100% CNT100
2: J P[ 2] 100% CNT100
3: L P[ 3] 200mm/sec FINE
4: Pickup ('Pieza2') From ('SoporteA') With ('GP: 1 - UT: 1 (PINZA)')
5: L P[ 4] 200mm/sec CNT100
6: J P[ 5] 100% CNT100
7: L P[ 6] 500mm/sec FINE
8: Drop ('Pieza2') From ('GP: 1 - UT: 1 (PINZA)') On ('Base')
9: L P[ 7] 500mm/sec FINE
10: J P[ 8] 100% CNT100 ACC110
11: WAIT 5.00(sec)
12: J P[ 9] 100% CNT100
13: L P[ 10] 500mm/sec FINE
14: WAIT 0.50(sec)
15: Pickup ('CAJA') From ('Base') With ('GP: 1 - UT: 1 (PINZA)')
16: WAIT 0.50(sec)
17: J P[ 11] 100% CNT100
18: J P[ 12] 100% CNT100 ACC150
19: L P[ 13] 200mm/sec FINE
20: WAIT 0.50(sec)
21: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('BaseA')
22: WAIT 0.50(sec)
23: L P[ 14] 200mm/sec FINE
24: J P[ 15] 60% CNT100
25: WAIT 2.00(sec)
26: J P[ 16] 60% CNT100
27: L P[ 17] 200mm/sec FINE
```

28: WAIT 0.50(sec)
29: Pickup ('CAJA') From ('BaseA') With ('GP: 1 - UT: 1 (PINZA)')
30: WAIT 0.50(sec)
31: L P[18] 200mm/sec FINE
32: L P[19] 200mm/sec FINE
33: J P[20] 100% CNT100
34: L P[21] 200mm/sec FINE
35: Drop ('CAJA') From ('GP: 1 - UT: 1 (PINZA)') On ('Base')
36: L P[22] 500mm/sec FINE
37: J P[23] 100% CNT100
38: L P[24] 200mm/sec FINE
39: Pickup ('Pieza1') From ('Base') With ('GP: 1 - UT: 1 (PINZA)')
40: L P[25] 400mm/sec FINE
41: J P[26] 100% CNT100
42: L P[27] 200mm/sec CNT100
43: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('BaseA')
44: L P[28] 200mm/sec FINE
45: J P[29] 100% CNT100
46: WAIT 2.00(sec)
47: J P[30] 100% CNT100
48: L P[31] 200mm/sec FINE
49: Pickup ('Pieza1') From ('BaseA') With ('GP: 1 - UT: 1 (PINZA)')
50: L P[32] 200mm/sec FINE
51: J P[33] 100% CNT100
52: L P[34] 400mm/sec FINE
53: Drop ('Pieza1') From ('GP: 1 - UT: 1 (PINZA)') On ('Base')
54: L P[35] 400mm/sec CNT100
55: J P[36] 100% CNT100

56: L P[37] 400mm/sec FINE
57: Pickup ('Pieza2') From ('Base') With ('GP: 1 - UT: 1 (PINZA)')
58: L P[38] 400mm/sec FINE
59: J P[39] 100% CNT100
60: L P[40] 200mm/sec FINE
61: Drop ('Pieza2') From ('GP: 1 - UT: 1 (PINZA)') On ('SoporteA')
62: L P[41] 200mm/sec FINE
63: J P[42] 100% CNT100

10. Agradecimientos.

Este trabajo final de carrera no habría sido posible sin el apoyo de unas cuantas personas, a las que me gustaría agradecer y dedicar unas breves líneas.

Me gustaría agradecer a Josep Ribó Pablo todo su apoyo en la realización de este trabajo. Su dirección ha sido imprescindible para completarlo, además de asesorarme en todo momento con su experiencia y su conocimiento.

También querría agradecer, en general, a todos los profesores de la EPS que con sus clases han aportado su granito de arena para permitirme haber llegado hasta aquí.

Me gustaría agradecer especialmente a mi familia el apoyo incondicional que me han ofrecido desde siempre. A mis padres, a mi hermano, a mis abuelos, tíos y primos, gracias por vuestro apoyo, cariño y confianza.

Y no puedo dejar en el tintero a todos aquellos amigos que me han apoyado y me han animado durante este tiempo, en el que he estado trabajando y haciendo el TFC.

A todos vosotros, GRACIAS.